Ant Colony Optimization

Chapter · February 2010

DOI: 10.5772/9389 · Source: InTech

Ben-Lian Xu
Changshu Institute of Technology

14

Ant Colony Optimization

Benlian Xu, Jihong Zhu and Qinlan Chen
Changshu Institute of Technology
China

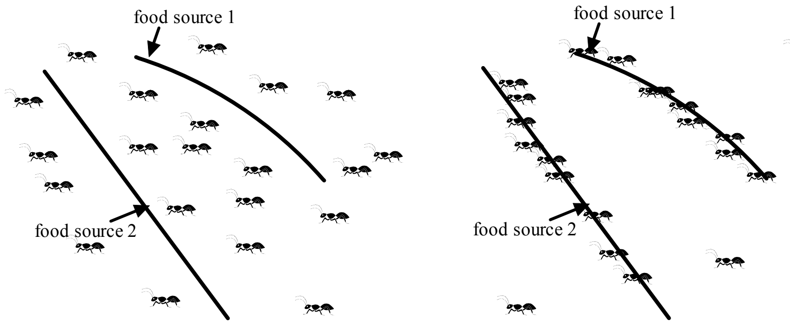
1. Introduction

Swarm intelligence is a relatively novel 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 ant colony optimization.

Ant colony optimization (ACO) algorithm, a novel population-based and meta-heuristic approach, was recently proposed by Dorigo et al. to solve several discrete optimization problems (Dorigo, 1996, 1997). The general ACO algorithm mimics the way real ants find the shortest route between a food source and their nest. The ants communicate with one another by means of pheromone trails and exchange information indirectly about which path should be followed. Paths with higher pheromone levels will more likely be chosen and thus reinforced later, while the pheromone intensity of paths that are not chosen is decreased by evaporation. This form of indirect communication is known as stigmergy, and provides the ant colony shortest-path finding capabilities. The first algorithm following the principles of the ACO meta-heuristic is the Ant System (AS) (Dorigo,1996), where ants iteratively construct solutions and add pheromone to the paths corresponding to these solutions. Path selection is a stochastic procedure based on two parameters, the pheromone and heuristic values, which will be detailed in the following section in this chapter. The pheromone value gives an indication of the number of ants that chose the trail recently, while the heuristic value is problem-dependent and it has different forms for different cases.

Due to the fact that the general ACO can be easily extended to deal with other optimization problems, its several variants has been proposed as well, such as Ant Colony System (Dorigo,1997), rank-based Ant System (Bullnheimer,1999), and Elitist Ant System (Dorigo,1996) . And the above variants of ACO have been applied to a variety of different problems, such as vehicle routing (Montemanni,2005), scheduling (Blum,2005), and travelling salesman problem (Stützle,2000). Recently, ants have also entered the data mining domain, addressing both the clustering (Kanade,2007), and classification task (Martens et al.,2007).

This chapter will focus on another application of ACO to track initiation in the target tracking field. To the best of our knowledge, there are few reports on the track initiation using the ACO. But in the real world, it is observed that there is a case in which almost all ants are inclined to gather around the food sources in the form of line or curve. Fig. 1 shows the evolution process of ants searching for foods. Initially, all ants are distributed randomly in the plane as in Fig.1 (a), and a few hours later we find that most of ants gather together around the food sources as shown in Fig.1 (b). Taking inspiration from such phenomenon, we may regard these linear or curvy food sources as tentative tracks to be initialized, and the corresponding ant model is established from the optimal aspect to solve the problem of multiple track initiation.



(a) Initial distribution of ants (b) The distribution of ants a few hours later

Fig. 1. The evolution process of ant search for foods

The remainder of this chapter is structured as follows. First, in section 2, the widely used ant system and its successors are introduced. Section 3 gives the new application of ACO to the track initiation problem, and the system of ants of different tasks is modeled to coincide with the problem. The performance comparison of ACO-based techniques for track initiation is carried out and analysized in Section 4. Finally, some conclusions are drawn.

2. Ant System and Its Direct Successors

2.1 Ant System

Initially, three different versions of AS were developed (Dorigo et al., 1991), namely ant-density, ant-quantity, and ant-cycle. In the ant-density and ant-quantity versions the ants updated the pheromone directly after a move from one city to an adjacent city, while in the ant-cycle version the pheromone update was only done after all the ants had constructed the tours and the amount of pheromone deposited by each ant was set to be a function of the tour quality.

The two main phases of the AS algorithm constitute the ants' solution construction and the pheromone update. In AS, a good way to initialize the pheromone trails is to set them to a value slightly higher than the expected amount of pheromone deposited by the ants in one iteration. The reason for this choice is that if the initial pheromone values are too low, then the search is quickly biased by the first tours generated by the ants, which in general leads toward the exploration of inferior zones of the search space. On the other side, if the initial pheromone values are too high, then many iterations are lost waiting until pheromone evaporation reduces enough pheromone values, so that pheromone added by ants can start to bias the search.

Tour Construction

In AS, m (artificial) ants incrementally build a tour of the TSP. Initially, ants are put on randomly chosen cities. At each construction step, ant k applies a probabilistic action choice rule, called random proportional rule, to decide which city to visit next. In particular, the probability with which ant k, located at city i, chooses to go to city j is

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}]^{\alpha} \cdot [\eta_{il}]^{\beta}}, \quad if\ j \in N_i^k, \tag{1}$$

where $\eta_{ij} = 1/d_{ij}$ is a heuristic value that is computed in advance, α and β are two parameters which determine the relative importance of the pheromone trail and the heuristic information, and N_i^k is the set of cities that ant k has not visited so far. By this probabilistic rule, the probability of choosing the arc (i, j) may increase with the bigger value of the associated pheromone trail τ_{ij} and of the heuristic information value η_{ij}. The role of the parameters α and β is described as below. If $\alpha = 0$, the closest cities are more likely to be selected: this corresponds to a classic stochastic greedy algorithm. If $\beta = 0$, it means that the pheromone is used alone, without any heuristic bias. This generally leads to rather poor results and, in particular, for values of $\alpha > 1$ it leads to earlier stagnation situation, that is, a situation in which all the ants follow the same path and construct the same tour, which, in general, is strongly suboptimal.

Each ant maintains a memory which records the cities already visited. And moreover, this memory is used to define the feasible neighbourhood N_i^k in the construction rule given by equation (1). In addition, such a memory allows ant k both to compute the length of the tour T^k it generated and to retrace the path to deposit pheromone for upcoming global pheromone update.

Concerning solution construction, there are two different ways of implementing it: parallel and sequential solution construction. In the parallel implementation, at each construction step all ants move from their current city to the next one, while in the sequential implementation an ant builds a complete tour before the next one starts to build another one. In the AS case, both choices for the implementation of the tour construction are equivalent in the sense that they do not significantly influence the algorithm's behaviour.

Update of Pheromone Trails

After all the ants have constructed their tours, the pheromone trails are updated. This is done by first lowering the pheromone value on all arcs by a factor, and then adding an amount of pheromone on the arcs the ants have crossed in their tours. Pheromone evaporation is implemented by the following law

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}, \quad \forall (i, j) \in L \tag{2}$$

where $0 < \rho < 1$ is the pheromone evaporation rate. The parameter ρ is used to avoid unlimited accumulation of the pheromone trails and it enables the algorithm to "forget" bad decisions previously taken. In fact, if an arc is not chosen by the ants, its associated pheromone value decreases exponentially with the number of iterations. After evaporation, all ants deposit pheromone on the arcs they have crossed in their tour:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k}, \forall(i,j)\in L \tag{3}$$

where $\Delta\tau_{ij}^{k}$ is the amount of pheromone ant k deposits on the arcs it has visited. It is defined as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/C^{k} & if\ arc\ (i,j)\ belongs\ to\ T^{k} \\ 0 & otherw \end{cases} \tag{4}$$

where C^{k}, the length of the tour T^{k} travelled by ant k, is computed as the sum of the lengths of the arcs belonging to T^{k}. By means of equation (4), the shorter an ant's tour is, the more pheromone the arcs belonging to this tour receive. In general, arcs that are used by many ants and which are part of short tours, receive more pheromone and are, therefore, more likely to be chosen by ants in the following iterations of the algorithm.

2.2. Elitist Ant System

The elitist strategy for Ant System (EAS) (Dorigo,1996) is, in principle, to provide a strong additional reinforcement to the arcs belonging to the best tour found since the start of the algorithm. Note that this additional feedback to the best-so-far tour is another example of a daemon action of the ACO meta-heuristics.

Update of Pheromone Trails

The additional reinforcement of tour T^{bs} is achieved by adding a quantity e/C^{bs} to its arcs, where e is a parameter that defines the weight given to the best-so-far tour T^{bs}, and C^{bs} is its length. Thus, equation (3) for the pheromone deposit becomes

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} + e\Delta\tau_{ij}^{bs} \tag{5}$$

where $\Delta\tau_{ij}^{k}$ is defined as in equation (4) and $\Delta\tau_{ij}^{bs}$ is defined as follows:

$$\Delta\tau_{ij}^{bs} = \begin{cases} 1/C^{bs} & if\ arc\ (i,j)\ belongs\ to\ T^{bs} \\ 0 & otherw \end{cases} \tag{6}$$

Note that in EAS the pheromone evaporation is implemented as in AS.

2.3. Rank-Based Ant System

Another improvement over AS (Bullnheimer,1999) is the rank-based version of AS (AS_{rank}). In AS_{rank} each ant deposits an amount of pheromone that decreases with its rank. Additionally, as in EAS, the best-so-far ant always receives the largest amount of pheromone in each iteration.

Update of Pheromone Trails

Before updating the pheromone trails, the ants are sorted by increasing tour length and the quantity of pheromone an ant deposits is weighted according to the rank of the ant. In each iteration, assume that total W best-ranked ants are considered, and only the $(W-1)$ best-ranked ants and the ant that produced the best-so-far tour are allowed to deposit pheromone. The best-so-far tour gives the strongest feedback with weight w; the r th best ant of the current iteration contributes to pheromone updating with the value $1/C^r$ multiplied by a weight given by max $\{0, W-r\}$. Thus, the AS_{rank} pheromone update rule is

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{W-1} (W-r)\Delta\tau_{ij}^{r} + w\Delta\tau_{ij}^{bs} \tag{7}$$

where $\Delta\tau_{ij}^{r} = 1/C^r$ and $\Delta\tau_{ij}^{bs} = 1/C^{bs}$.

2.4 *Max- Min* Ant System

Max-Min Ant System (MMAS) (Stützle & Hoos, 2000) introduces some main modifications with respect to AS. First, it strongly exploits the best tours found: only either the iteration-best ant, that is, the ant that produced the best tour in the current iteration, or the best-so-far ant is allowed to deposit pheromone. Unfortunately, such a strategy may lead to a stagnation situation in which all ants follow the same tour, because of the excessive growth of pheromone trails on arcs of a good, although suboptimal, tour. To counteract this effect, a second modification introduced by MMAS is that it limits the possible range of pheromone trail values to the interval $[\tau_{\min}, \tau_{\max}]$. Second, the pheromone trails are initialized to the upper pheromone trail limit, which, together with a small pheromone evaporation rate, increases the exploration of tours at the start of the search. Finally, in MMAS, pheromone trails are reinitialized each time the system approaches stagnation or when no improved tour has been generated for a certain number of consecutive iterations.

Update of Pheromone Trails

After all ants have constructed a tour, pheromones are updated by applying evaporation as in AS, followed by the deposit of new pheromone as follows:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{best}, \tag{8}$$

where $\Delta\tau_{ij}^{best} = 1/C^{best}$. The ant which is allowed to add pheromone may be either the best-so-far, in which case $\Delta\tau_{ij}^{best} = 1/C^{bs}$, or the iteration-best, in which case $\Delta\tau_{ij}^{best} = 1/C^{ib}$, where C^{ib} is the length of the iteration-best tour. In general, in MMAS implementations both the iteration-best and the best-so-far update rules are used in an alternate way. Obviously, the choice of the relative frequency with which the two pheromone update rules are applied has an influence on how greedy the search is: When pheromone updates are always performed by the best-so-far ant, the search focuses very quickly around T^{bs}, whereas when it is the iteration-best ant that updates pheromones, then the number of arcs that receive pheromone is larger and the search is less directed.

Pheromone Trail Limits

In MMAS, lower and upper limits $\tau_{\min}$ and $\tau_{\max}$ on the possible pheromone values on any arc are imposed in order to avoid earlier searching stagnation. In particular, the imposed pheromone trail limits have the effect of limiting the probability p_{ij} of selecting a city j when an ant is in city i to the interval $[p_{\min}, p_{\max}]$, with $0 < p_{\min} \le p_{ij} \le p_{\max} \le 1$. Only when an ant k has just one single possible choice for the next city, that is $\left|N_i^k\right| = 1$, we have $p_{\min} = p_{\max} = 1$.

It is easy to show that, in the long run, the upper pheromone trail limit on any arc is bounded by $1/\rho C^*$, where C^* is the length of the optimal tour. Based on this result, MMAS uses an estimate of this value, $1/\rho C^{bs}$, to define $\tau_{\max}$: each time a new best-so-far tour is found, the value of $\tau_{\max}$ is updated. The lower pheromone trail limit is set to $\tau_{\min} = \tau_{\max} / \kappa$, where κ is a parameter (Stützle & Hoos, 2000).

Pheromone Trail Initialization and Re-initialization

At the start of the algorithm, the initial pheromone trails are set to an estimate of the upper pheromone trail limit. This way of initializing the pheromone trails, in combination with a small pheromone evaporation parameter, causes a slow increase in the relative difference in the pheromone trail levels, so that the initial search phase of MMAS is very explorative.

Note that, in MMAS, pheromone trails are occasionally re-initialized. Pheromone trail re-initialization is typically triggered when the algorithm approaches the stagnation behaviour or if for a given number of algorithm iterations no improved tour is found.

3. ACO for Track Initiation of Bearings-only multi-target tracking

3.1 Problem Presentation

Bearings-only multi-target tracking (BO-MTT) (Nardone, 1984 ; Dogancay, 2004, 2005) in a bistatic system can be described as: given a time history of noise-corrupted bearing measurements from two observers, the objective is to obtain optimum estimation of the positions, velocities and accelerations of all targets. Generally, the whole process of target tracking includes track initiation, track maintenance and track deletion. To the best of our knowledge, however, many reported literature mainly focused on the track maintenance, i.e. target tracking, without considering the track initiation process, after the motion of each target is modelled. Actually, track initiation plays an important role in evaluating the performance of subsequent target tracking, and improperly initiated tracks may either lead to target loss or the increase of consumption of limited resources.

In the case of multi-sensor-multi-target BOT, for instance, two-sensor-two-target BOT at a given scan, four Line of Sights (LOSs) are available alone to determine which LOS belongs to some target of interest. Usually, such a problem can also be dealt with the general track initiation techniques widely used in the radar tracking field through intersecting these LOSs to obtain a group of candidates of true targets' position points. However, such an operation will result in some intersections including both the true "target positions" and the virtual "target position" called "ghost", as shown in Fig.2. These "ghosts", in fact, do not belong to any target (denoted by position points 3 and 4). Due to this fact, the origin uncertainty of obtained position candidates should be discriminated and this issue forms the topic of this section. In addition, such a problem becomes harder to handle in the presence of clutter.

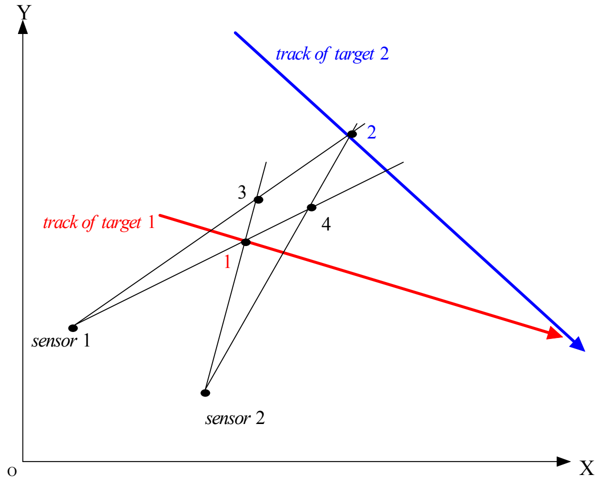


Fig. 2. The generated "ghosts" in case of two-sensor-two-target BOT

3.2 Motive

In the image detection field, the Hough transform (H-T) has been recognized as a robust technique for line or curve detection and also have been largely applied by scientific community (Bhattacharya, 2002; Shapiro, 2005). The basic idea of H-T is to transform a point (x, y) in the Cartesian coordinate system onto a curve in the (ρ, θ) parameter space, which is formulated as

$$\rho = x\cos\theta + y\sin\theta \tag{9}$$

where ρ is the distance from the line through (x, y) to the origin, and θ is the angle to the normal with the x axis. The angle θ varies from 0^0 to 180^0, while the ρ may be either positive or negative.

So, it is observed that, if a set of points in the Cartesian coordinate lie on the same line, all curves each corresponding to a point must intersect at a same point denoted by (ρ_0, θ_0) in the parameter space. Inspired by this phenomenon, the H-T technique can be utilized to initialize the track of target which makes a uniform rectilinear motion.

3.3 Solution to Multi-Target Track Initiation by ACO

In this section, we will investigate the problem of multi-target track initiation. First, a objective function is presented to describe the property of the multi-target track initiation. Second, a novel ACO algorithm, called different tasks of ants, is modelled to initiate the tracks of interest.

As noted before, if there are n curves in the parameter space, at most C_n^2 intersections are obtained in general. However, in a real tracking scenario, these curves will not strictly intersect the point but several points distributed in the parameter space due to the existence

of measurement error. Even so, these points are still distributed in a small region, and thus such a small area could be deemed as an objective function to be optimized.
For the case of two given tracks, the corresponding intersections in the parameter space are plotted in Fig.3, and for the upper left expanded subfigure, which corresponds to target 1, the minimum and maximum values of θ could be obtained and then denoted by $\theta_{\min}$ and $\theta_{\max}$, respectively. Similarly, the related minimum and the maximum values of ρ are also found and denoted by $\rho_{\min}$ and $\rho_{\max}$, respectively.

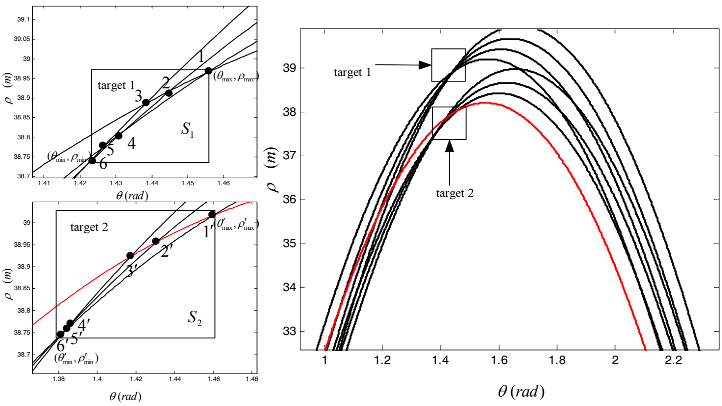


Fig. 3. A case of determination of objective function in the parameter space
As a result, two rectangular blocks are formed and the area of each is calculated as

$$S_i \triangleq (\theta^i_{\max} - \theta^i_{\min}) \bullet (\rho^i_{\max} - \rho^i_{\min}), \tag{10}$$

and the objective function J is defined as

$$J = \min \sum_{r=1}^{M} S_{r(r_1 - r_2 - r_3 - r_4)}$$
$$s.t \quad r_k \neq m_k \qquad \forall r_1 - r_2 - r_3 - r_4, m_1 - m_2 - m_3 - m_4 \in \Theta, \tag{11}$$
$$k = 1, \ldots, 4;$$

where $r_1 - r_2 - r_3 - r_4$ or $m_1 - m_2 - m_3 - m_4$ is the possible track in the track space Θ, M is the number of tracks to be initialized.
Afterwards, the ants of different tasks will be investigated, and it has the following characteristics:

1) The number of tasks is equal to the one of tracks to be initiated, or equal to the one of targets of interest.
2) The traditional ACO algorithm builds solutions in an incremental way, but the proposed system of different tasks of ants builds solutions in parallel way. Especially, in the proposed system of ants of different tasks, the thought of both collaboration and competition between ants is considered and introduced. For instance, ants of the same task search for foods in a collaborative way, while ants of different tasks will compete with each other during establishing solutions.

3) Ants of the same task are dedicated to finding their best solution, and a set of all best solutions found by ants of different tasks constitute the solutions to Eq. (11) we describe.
4) In the system of ants of different tasks, the search space depends not only on the measurement returns at the next scan but also on the prior knowledge of target motion.

The determination of search space

In the case of bearings-only two-sensor- M -target tracking, the sampling data of the first four scans are utilized sequentially to initiate tracks, and then total four search spaces, i.e., Ω_1, Ω_2, Ω_3, and Ω_4, are obtained sequentially. Suppose that the prior knowledge about target motion, such as the minimum and maximum velocities denoted by $v_{\min}$ and $v_{\max}$ respectively, is known and then utilized to construct an annular region whose inner and outer radiuses are determined by $r_1 = \| \boldsymbol{v}_{\min} \| \bullet T$ and $r_2 = \| \boldsymbol{v}_{\max} \| \bullet T$, respectively, where T denotes the sampling interval. For instance, if an ant is now located at position i in Ω_1, then the ant will visit the next position located in the shadow section covered by both the annular region and Ω_2, which is denoted by $\tilde{\Omega}_2^i$ in Fig.4.

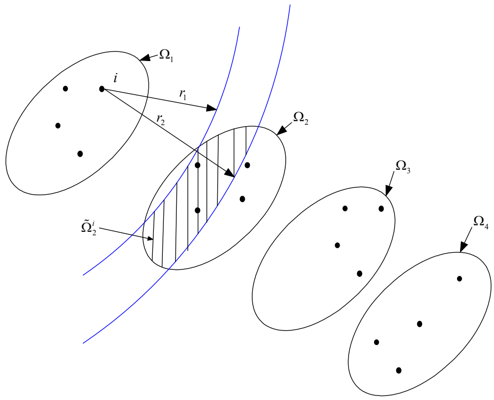


Fig. 4. The determination of search spaces

Track Candidate Construction Using the Ants of Different Tasks

Initially, $\tilde{M}$ ants of different tasks are placed randomly on position candidates in the first search space Ω_1, then each ant of a given task visits probabilistically the position candidate in the next search space. Suppose that an ant of a given task s is now located at position i in $\tilde{\Omega}_{\tilde{i}}^i$ ($1 \le \tilde{i} \le 3$), then the ant will visit position j in the next search space by applying the following probabilistic formula:

$$j = \begin{cases} \arg\max_{j \in \tilde{\Omega}^i_{\tilde{i}+1}} \left\{ \left[\frac{\tau^s_{i,j}}{\tau_{i,j}} \right]^\alpha \bullet \left[\eta_{i,j} \right]^\beta \bullet \left[\frac{1}{\tau_{i,j} - \tau^s_{i,j}} \right]^\gamma \right\} & \text{if } q \le q_0 \\ \bar{J} & \text{otherwise} \end{cases}, \tag{12}$$

and $\bar{J}$ is a random variable selected according to the following probability distribution

$$P(j) = \begin{cases} \dfrac{\left[\frac{\tau^s_{i,j}}{\tau_{i,j}} \right]^\alpha \bullet \left[\eta_{i,j} \right]^\beta \bullet \left[\frac{1}{\tau_{i,j} - \tau^s_{i,j}} \right]^\gamma}{\sum_{l \in \tilde{\Omega}^i_{\tilde{i}+1}} \left[\frac{\tau^s_{i,l}}{\tau_{i,l}} \right]^\alpha \bullet \left[\eta_{i,l} \right]^\beta \bullet \left[\frac{1}{\tau_{i,l} - \tau^s_{i,l}} \right]^\gamma} & \text{if } j \in \tilde{\Omega}^i_{\tilde{i}+1} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where $\tau^s_{i,j}$ denotes the pheromone amount deposited by ants of task s on trail (i, j), $\tau_{i,j}$ is the total pheromone amount deposited by all ants of different tasks on trail (i, j), γ shows the repulsion on the foreign pheromones left on the trail (i, j), q is a random number uniformly distributed between 0 and 1, and q_0 is a parameter which determines the relative importance of the exploitation of good solutions versus the exploration of search spaces.

According to the search spaces discussed above, Fig. 5 plots the process of how the heuristic value is calculated from search spaces Ω_1 to Ω_2, namely, if an ant will move from positions i to j, the corresponding heuristic value can be defined as

$$\eta_{i,j} = \exp\left(-\frac{(d_{i,j} - r_0)^2}{2(r_2 - r_1)^2} \right), \tag{14}$$

where $d_{i,j}$ denotes the distance between positions i and j, and r_0 is equal to $(r_2 - r_1)/2$. Note that if position j falls out of $\tilde{\Omega}^i_2$, we set $\eta_{i,j} = 0$, and the search failure is declared for the current ant.

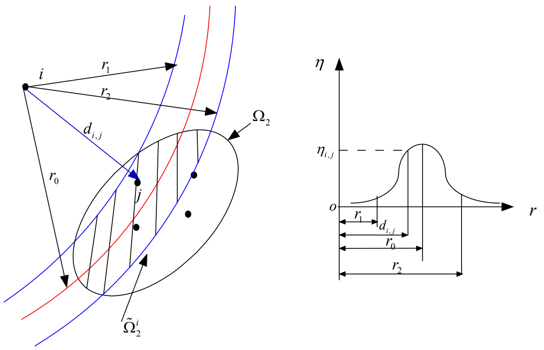


Fig. 5. The calculation of heuristic value

Update of Pheromone

The pheromone update is performed in two phases, namely, local update and global update. While building a solution, if an ant of task s carries out the transition from positions i to j, then the pheromone level of the corresponding trail is changed in the following way:

$$\tau_{i,j}^{s} \leftarrow (1-\rho)\tau_{i,j}^{s} + \tau_{0}^{s} \text{ ,} \tag{15}$$

where τ_0^s is the initial pheromone level of ants of task s.

Once all ants of different tasks at a given iteration have visited four candidate positions each from different sampling indices, the pheromone amount on each established track will be updated globally. Here, we use the best-so-far-solution found by ants of the same task, i.e. the best solution found from the start of the algorithm run, to update the corresponding pheromone trail. We adopt the following rule

$$\tau_{i,j}^{s} \leftarrow (1-\rho)\tau_{i,j}^{s} + \sum_{k=1}^{p} \Delta\tau_{i,j}^{s,k} \text{ .} \tag{16}$$

where $\Delta\tau_{i,j}^{s,k}$ is the pheromone amount that ant k of task s deposits on the trail (i, j) it has traveled at the current iteration, and p is the number of ants. In the case of bearings-only multi-sensor-multi-target tracking, $\Delta\tau_{i,j}^{s,k}$ is set to a constant.

4. A Comparison of ACO-Based Methods for Track Initiation

4.1 The Problem

Two cases are investigated here, namely two and three tracks' initiation problems. For each scenario, the performance of track initiation is investigated both in clutter-free environments and in clutter environments, respectively.

Two fixed sensors used to measure the targets' bearings are located at $(0,0)$ and $(18km,0)$ respectively in a surveillance region. The standard deviation of the bearing measurements for each sensor is taken as 0.1^0, and the sampling interval is set to be $T=10s$. The case in which each target makes a uniform rectilinear motion is considered, and the initial state of each target is illustrated in Table 1.

Scenarios	Targets	x (km)	y (km)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)
1	1	60	30	50	-100
	2	80	60	150	-150
2	1	60	30	50	-100
	2	80	60	150	-150
	3	60	50	80	-120

Table 1. The initial position and velocity of each target in the two considered scenarios

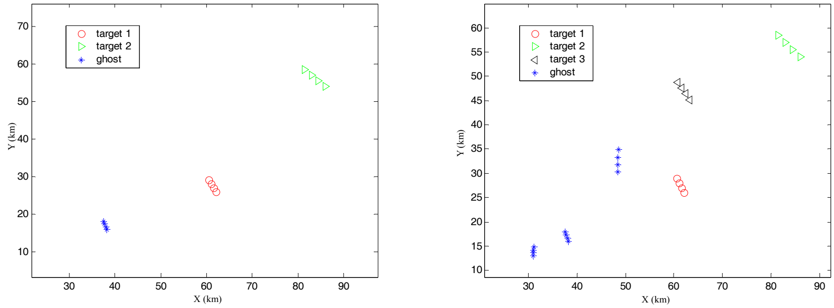


Fig. 6. The target position candidates in a "clutter-free" environment (left: Scenario 1, right: Scenario 2)

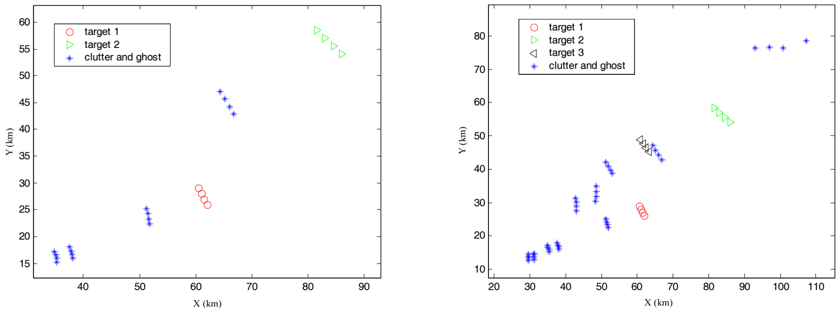


Fig. 7. The target position candidates in clutter environments (left: Scenario 1, right: Scenario 2)

Figs.6 and 7 depict a part of position candidates obtained by intersecting LOSs at each scan, and our object is to discriminate the true "positions" of each target of interest. Here, we use two ACO-based techniques, namely the Ant System (called the traditional ACO) and the system of ants of different tasks (called the proposed ACO).

Other parameters related to the two ACO-based methods are illustrated in Table 2

Parameter	Value	Parameter	Value
α	0.01	$\Delta\tau$	0.03
β	0.2	$\tilde{M}$	$3M^2$
γ	2	$\lvert v_{\min} \rvert$	$100m/s$
ξ	0.8	$\lvert v_{\max} \rvert$	$400m/s$
q_0	0.7	$\lvert a_{\max} \rvert$	$15m/s^2$
τ_0	0.05	N_θ	50

Table 2. The Parameter Settings for ACO-related Methods

4.2 Evaluation Indices

Two performance indices are introduced to evaluate the system of ants of different tasks, i.e.

The probability of false track initiation: assuming $\tilde{N}$ Monte-Carlo runs are performed, we define the probability of false track initiation as

$$F \triangleq \sum_{i=1}^{\tilde{N}} f_i \Big/ \sum_{i=1}^{\tilde{N}} n_i \,, \tag{17}$$

where f_i denotes the number of false initiated tracks at the i th Monte-Carlo run, and n_i is the total number of initiated tracks.

The probability of correct initiation of at least j tracks: if at least j ($1 \le j \le M$) tracks are initiated correctly, its corresponding probability is

$$C_j \triangleq \sum_{i=1}^{\tilde{N}} l_{ij} \Big/ \tilde{N} \,, \tag{18}$$

where l_{ij} is a binary variable and defined as

$$l_{ij} = \begin{cases} 1 & \textit{if at least j tracks are initiated correctly} \\ 0 & \textit{otherwise} \end{cases} \tag{19}$$

at the i th Monte-Carlo run.

4.3 Results

All results in Tables 3 to 6 are averaged over 10,000 Monte-Carlo runs. According to the evaluation indices we introduce, the traditional ACO algorithm performs as well as the proposed one, as illustrated in Tables 3 and 4, in clutter-free environments. However, in the presence of clutter, the proposed ACO algorithm shows a significant improvement over the traditional one with respect to the probability of false track initiation, as shown in Tables 5 and 6.

Evaluation indices		The traditional ACO	The proposed ACO
Pro. of false track initiation (F)		0.0001	0.0002
Pro. of correct initiation of at least j tracks(C_j)	C_1	1.0000	1.0000
	C_2	0.9998	0.9997

Table 3. Performance comparison for two-track-initiation problem in clutter-free environments

Evaluation indices		The traditional ACO	The proposed ACO
Pro. of false track initiation (F)		0.0048	0.0046
Pro. of correct initiation of at least j tracks(C_j)	C_1	1.0000	1.0000
	C_2	1.0000	1.0000
	C_3	0.9857	0.9861

Table 4. Performance comparison for three-track-initiation problem in clutter-free environments

Evaluation indices		The traditional ACO	The proposed ACO
Pro. of false track initiation (F)		0.0348	0.0107
Pro. of correct initiation of at least j tracks(C_j)	C_1	1.0000	1.0000
	C_2	0.9787	0.9997

Table 5. Performance comparison for two-track-initiation problem in clutter environments

Evaluation indices		The traditional ACO	The proposed ACO
Pro. of false track initiation (F)		0.0672	0.0380
Pro. of correct initiation of at least j tracks(C_j)	C_1	1.0000	1.0000
	C_2	0.9594	1.0000
	C_3	0.9267	0.9861

Table 6. Performance comparison for three-track-initiation problem in clutter environments

Among 10,000 Monte-Carlo runs, only the cases of all tracks being initiated successfully are investigated and called effective runs later. For the objectivity of comparison, we select the worst case, in which the maximum running time for each ACO algorithm is evaluated, from the effective runs.

Fig. 8 depicts the trends of objective function evolution with the increasing number of iterations in scenario 2. Compared with the traditional ACO algorithm, the proposed one requires fewer iterations for convergence in clutter-free or clutter environments. According to Tables 3 and 4, although the performance of the traditional ACO algorithm is comparable to that of the proposed one, we find that the proposed ACO one seems more practical due to less running time needed. Figs. 9 and 10 depict varying curves of pheromone on the true targets' tracks, it is observed that the amount of pheromone on each "true" track increases in a moderate way, which means most ants prefer choosing these tracks and regarded them as optimal solutions.

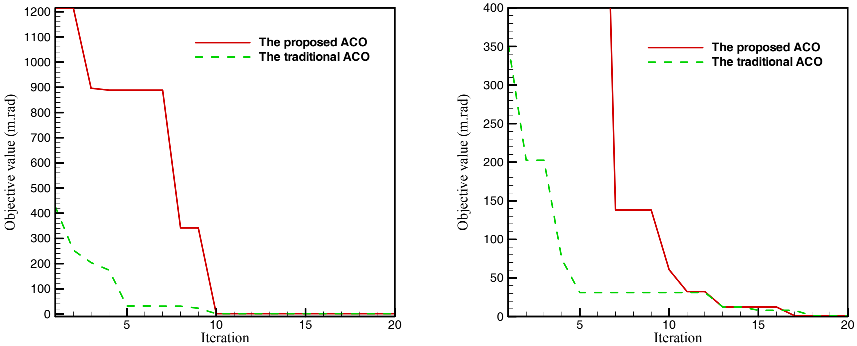


Fig. 8. Objective function curves (left: In clutter-free environments; right: In clutter environments)

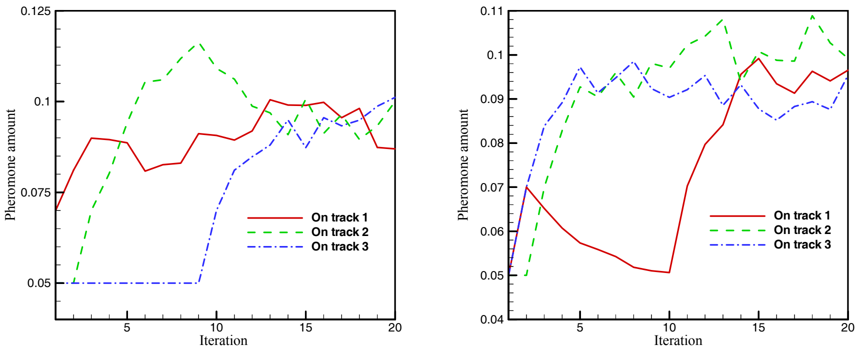


Fig. 9. Pheromone curves in clutter-free environments (left: The proposed ACO; right: The traditional ACO)

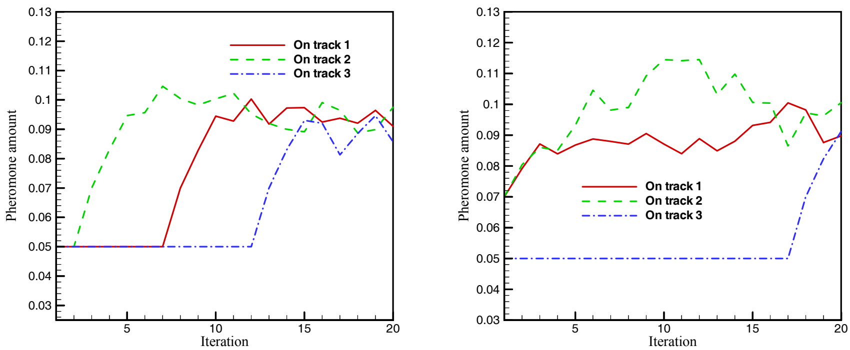


Fig. 10. Pheromone curves in clutter environments (left: The proposed ACO; right: The traditional ACO)

5. Conclusion

This chapter mainly aims to introduce some widely used ACO algorithms and their origins, such as the AS, EAS, MMAS, and so on. It is found that all concerns are focused on the pheromone update strategy. Some uses the best-so-far-ant or the iteration-best ant independently/interactively to update the trail that ants travelled. Meanwhile, the update law may differ a bit for different ACO algorithms. Among the four ACO algorithms, two versions have received great popularities in various applications, i.e. AS and MMAS. Another contribution in this chapter is the extension of the general ACO algorithm to the system of ants of different tasks, and its behaviour is modelled and implemented in the track initiation problems. Simulation results are also presented to show the effectiveness of the novel ACO algorithm. According to the example presented in this chapter, we believe that the general framework of AS can be modified to solve various optimal or non-optimal problems.

6. References

B. Bullnheimer; R. F. Hartl & C. Strauss. (1999). A new rank based version of the ant system: A computational study, *Central Eur. J. Oper. Res. Econ.*, Vol. 7, No. 1, 25–38, ISSN 1435-246X.

C. Blum. (2005). Beam-ACO—hybridizing ant colony optimization with beam search: An application to open shop scheduling, *Comput. Oper. Res.*, Vol. 32, No. 6, 1565–1591, ISSN 0305-0548.

David Martens; Manu De Backer & Raf Haesen. (2007). Classification With Ant Colony Optimization, *IEEE Trans. on Evolutional Computation*, Vol. 11, No. 5, October 2007, 651-665, ISSN 1089-778X.

Kutluyll Dogancay. (2004). On the bias of linear least squares algorithm for passive target localization, Signal Processing, Vol. 84, No. 3, 475-486, ISSN 0165-1684.

Kutluyll Dogancay. (2005). Bearings-only target localization using total least squares, *Signal Processing*, Vol. 85, No. 9,1695-1710, ISSN 0165-1684.

M. Dorigo; V. Maniezzo & A. Colorni. (1991). Positive Feedback as a Search Strategy, Technical Report 91–016, Politecnico di Milano, Milano, Italy.

M. Dorigo; V. Maniezzo & A. Colorni. (1996). The ant system: optimization by a colony of cooperating agents, *IEEE Trans. on System, Man, and Cybernetics-part B,* Vol.26, No. 1, 29-42, ISSN 1083-4419.

M. Dorigo & L. M. Gambardella. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Trans. on Evolutional Computation*, Vol.1, No. 1, 53-66, ISSN 1089-778X.

P. Bhattacharya; A. Rosenfeld & I. Weiss. (2002). Point-to-line mappings as Hough transforms, *Pattern Recognition Letters*, Vol. 23, No. 4, 1705-1710, ISSN 0167-8655.

Parag M. Kanade & Lawrence O. Hall. (2007). Fuzzy Ants and Clustering, *IEEE Trans. on System, Man, and Cybernetics-part A,* Vol. 37, No. 5, September 2007, 758-769,ISSN 1083-4427.

R. Montemanni; L. M. Gambardella; A. E. Rizzoli & A. Donati. (2005). Ant colony system for a dynamic vehicle routing problem, *J. Combinatorial Optim.*, Vol. 10, No. 4, 327–343, ISSN 1573-2886.

S.C. Nardone; A.G. Lindgren & K.F. Gong. (1984). Fundamental properties and performance of conventional bearings-only target motion analysis, *IEEE Transactions on Aerospace* and Electronic Systems, Vol. 29, No. 9, 775-787, ISSN 0018-9251.

T. Stützle & H. H. Hoos. (2000). $\mathcal{MAX}$-$\mathcal{MIN}$ ant system, *Future generation computer systems,* Vol.16, 889-914, ISSN 0167-739X.

V. Shapiro. (2006). Accuracy of the straight line Hough transform: the non-voting approach, *Computer Vision and Image Understanding*, Vol. 103, No. 1, 1-21, ISSN 1077-3142.