

## Review of Ant Colony Optimization Model for Suspended Sediment Estimation

Omolbani Mohamad Rezapour-Faculty of Engineering University Putra Malaysia  
Amirahmad deghani - Gorgan University of Agricultural Sciences and Natural Resources  
Lee Teang Shui – Faculty of Engineering University Putra Malaysia

**Abstract:** Estimation of sediment concentration in rivers is very important for water resource projects planning and managements. The sediment concentration is generally determined from the direct measurement of sediment concentration of river or from sediment transport equations. Direct measurement is very expensive and cannot be conducted for all river gauge stations. However, sediment transport equations do not agree with each other and require many detailed data on the flow and sediment characteristics. Various models have been developed so far to identify the relation between discharge and sediment load. Most of the models based on regression method have some restrictive assumptions. Ant colony optimization (ACO) is now being used more frequently to solve optimization problems other than those for which they were originally developed. The main purpose of this paper is literature review of Ant Colony Optimization for suspended sediment estimation.

**Key word:** Suspended sediment, Ant colony optimization, River, Estimation.

### INTRODUCTION

Suspended sediment transport by rivers is an important phenomenon to both science and river engineering (Sichingabula, 1993). Correct estimation of sediment volume carried by a river is very important for many water resources projects. The prediction of river sediment load also constitutes an important issue in hydraulic and sanitary engineering (Aytek *et al.*, 2007).

Due to the high costs associated with the Suspended sediment transport by rivers much research over the last 30 years has been dedicated to the development of techniques to minimise the capital costs associated with such infrastructure. This process has been given the title of 'optimisation' or 'optimal design'.

Within the last decade, many researchers have shifted the focus of new optimisation from traditional optimisation techniques based on linear and non-linear programming to the implementation of heuristics derived from nature, namely; Genetic Algorithms (GAs), Simulated Annealing, the Shuffled Frog-Leaping Algorithm (SFLA), and Ant Colony Optimisation (ACO).

In the early 1990s, ant colony optimization (ACO) (Dorigo, 1992, Gambardella, 1997, Maniezzo, 1996) was introduced by M. Dorigo and colleagues as a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization (CO) problems. ACO belongs to the class of metaheuristics (Blum, 2003, Glover, 2002, Hoos, 2004), which are approximate algorithms used to obtain good enough solutions to hard CO problems in a reasonable amount of computation time.

Other examples of metaheuristics are tabu search (Glover *et al.* 1989, 1990, 1997), simulated annealing (Kirkpatrick, 1983, Cerny, 1985), and evolutionary computation (Holland, 1975, Rechenberg, 1973, Fogel, *et al.*, 1966). The inspiring source of ACO is the foraging behavior of real ants. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the ant deposits a chemical pheromone trail on the ground. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will guide other ants to the food source. As it has been shown in (Deneubourg *et al.*, 1990), indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources. This characteristic of real ant colonies is exploited in artificial ant colonies in order to solve CO problems.

#### **Biological Inspiration of Natural Ants:**

Ant Colony Optimization was originally inspired by the behaviour of natural ants. In the real world, ants release a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a

direction which is rich in pheromone. The famous bridge experiment (Deneubourg and Goss, 1989) shows that ants always can find the shortest path between the colony and food source. In the following illustration, we explain why ants are able to adjust to changes in the environment, such as new obstacles interrupting the shortest path.



**Fig. 1:** Ants' food path between their nest and the food source without obstacles

first, as shown in fig.1, ants follow the shortest path to carry food back to their colony. If there is an obstacle obstructs the path the ants will randomly choose another way around the obstruction (right, left, over or under). If we assume that the ants cannot go wider or over the obstruction, we can safely assume that approximately half of the ants will go right and the other half left when ants reach points A and B, as illustrated in fig.2. The ants that happen to pick the shorter path will obviously create a strong trail of pheromone a lot faster than the ants choosing the longer path. This will cause more and more ants to choose the shorter path until eventually all ants have found the shortest path as shown in fig.3.



**Fig. 2:** Ants explore new paths after encountering obstacle



**Fig. 3:** Ants find the new shortest path between the food and their nest

Ant Colony Optimization attempts to apply similar techniques in order to solve much more complex problems in real life. The main idea is to repeatedly use simulations of artificial ants as agents to generate new solutions to the problem at hand. The ant agents use information collected during past simulations to direct their next step in current run and this information is available from the modifiable medium (pheromone intensity on the trail). In general, the ACO approach attempts to solve an optimization problem by repeating the following two steps:

- Candidate solutions are constructed using a pheromone model, that is, a parametrized probability distribution over the solution space;
- The candidate solutions are used to modify the pheromone values in a way that is deemed to bias future sampling toward high quality solutions.

***Ant Colony Optimization Structure (ACO):***

Although they are inspired by natural ants, artificial ants in ACO have more capabilities and can solve more complicated problems. Artificial ants have memory to remember their path and can trace all the way back (in order to update the pheromone trail). They can compute the quality of the path and update the trail with a proportional pheromone amount while real ants can deposit more pheromone if more food is found at the food source.

Ant Colony Optimization is proposed as an approach for solving hard combinatorial optimization problems. It can be seen as an extension of the construction method using a special pheromone trail. In the implementation of many recently proposed ACO algorithms, local search is often added-in to improve the performance.

The basic framework of the ACO method remains the same. The ACO meta-Heuristic is as the pseudo code as below:

```

Ant Colony Optimization meta-heuristic
WHILE termination conditions not met DO
Schedule Activities
  Ant Based Solution Construction
  Pheromone update
Daemon Actions {optional}
END Schedule Activities
ENDWHILE
    
```

The Schedule Activities construction consists of three parts.

**Ant Based Solution Construction:** An ant constructively builds a solution to a problem by moving through the nodes of the construction graph  $G$  ( in which ants can construct solution by moving from component to component). Ants move by applying a stochastic local decision policy that makes use of the pheromone values and the heuristic values on components and/or connections of the construction graph. While moving, the ant remembers the partial solution it has built based on the path it was following on the construction graph.

***Pheromone Update:***

When adding a component to the current partial solution, an ant can update the values of the pheromone trails that were used for this construction step. This kind of pheromone update is called an online step-by-step pheromone update. Once an ant has built a solution, it can retrace the same path backwards by using its memory and update the pheromone trails of the used components and/or connections according to the quality of the solution it has built. This is called online delayed pheromone update. Another important concept in Ant Colony Optimization is pheromone evaporation. Pheromone evaporation is the process by which the pheromone trail intensity on the components decreases over time. From a practical point of view, pheromone evaporation is needed to avoid a too rapid convergence of the algorithm toward a sub-optimal region. It implements a useful form of forgetting, favouring the exploration of new areas in the search space.

***Daemons Actions:***

Daemon actions can be used to implement centralized actions which cannot be performed by single ants. Examples are the use of a local search procedure applied to the solutions built by the ants, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from a non-local perspective. As a practical example, the daemon can observe the path found by each ant in the colony and choose to deposit extra pheromone on the components used by the ant that built the best solution. Pheromone updates performed by the daemon are called offline pheromone updates.

The ACO procedure can be summarized as follows and different ACO algorithms are discussed subsequently:

**3.1 Ant System (AS)** In AS, solutions are constructed based on the transition probability,

$$P_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{l \in U} (\tau_{il})^\alpha (\eta_{il})^\beta}$$

where ' $\eta_{ij}$ ' is a local heuristic,  $\alpha$  and  $\beta$  are two parameters that determine the relative influence of the pheromone trail and the heuristic, and  $U$  denotes the set of candidate solutions to be chosen. The pheromone

trail can be updated as  $\tau_{ij}^{new} = \rho \tau_{ij}^{old} + \sum_{k=1}^{NA} \Delta \tau_{ij}^k$  where  $\rho$  is a parameter that controls the pheromone persistence ( $0 \leq \rho \leq 1$ ), i.e.,  $1-\rho$  represents the proportion of the pheromone evaporated, and  $NA$  is

number of ants, i.e., all ants can contribute to pheromone trail accumulation in the AS algorithm. Dorigo *et al.* (1991a, 1991b, 1996) propose three different approaches to find  $\Delta\tau_{ij}^k$  values for the TSP as follows:

- Ant Density:  $\Delta\tau_{ij}^k = Q$ ,
- Ant Quantity:  $\Delta\tau_{ij}^k = \frac{Q}{d_{ij}}$ ,
- Ant Cycle:  $\Delta\tau_{ij}^k = \frac{Q}{L^k}$ .

where Q denotes a constant related the quantity of pheromone trail ants laid,  $d_{ij}$  represents the distance between cities i and j, and  $L^k$  is the total tour length of the  $K^{th}$  ant. The ant density and ant quantity approaches deposit pheromone every time an ant goes from i to j, but the ant cycle deposits it only after a complete tour. Experiments indicate that ant cycle outperforms the other two approaches (Colomi *et al.* 1997, Dorigo *et al.* 1991a, 1991b, 1996).

### 3.2 AS<sup>elite</sup>:

The only difference between AS and AS<sup>elite</sup> is the pheromone update rule. In the AS algorithm, every ant has the same "weight" in contributing to the pheromone trail, but in AS<sup>elite</sup> the best ant contributes more than

other ants. Therefore, the pheromone update rule is revised to  $\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \sum_{k=1}^{NA} \Delta\tau_{ij}^k + \Delta\tau_{ij}^e$  where

$\Delta\tau_{ij}^e$  can be equal to, for example,  $e \cdot \frac{Q}{L^e}$ . In TSP, e denoting the number of the best ants used (elitist

ants), and  $L^e$  representing the solution of the best ant found so far (Bullnheimer *et al.* 1999b, Dorigo *et al.* 1991a, 1991b, 1996) (or AS-Reinfor in (Kaji 1999)).

### 3.3 AS<sup>rank</sup>:

Bullnheimer *et al.* propose an algorithm that enforces the pheromone trail by not only relying on the elitist ant but also some other "good" ants. The state transition rule is the same as the one in AS and AS<sup>rank</sup>. The contribution of an ant to the trail level update is weighted according to the rank, r, of the ant, and only the u best ants are considered, and u=e-1.

Therefore, the trail updating rule is as follows:  $\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \sum_{r=1}^{e-1} \Delta\tau_{ij}^r + \Delta\tau_{ij}^e$

where  $\Delta\tau_{ij}^r$ , for example in TSP, is equal to  $(e-r) \frac{Q}{L^e}$ . and  $\Delta\tau_{ij}^e = (e-r) \frac{Q}{L^e}$ .  $L^e$  represents

the solution of the best ant found so far (Bullnheimer *et al.* 1999b).

### 3.4 Ant-Q (Dorigo and Gambardella 1996, Gambardella and Dorigo 1995):

Gambardella *et al.* use the idea of Q-learning (Watkins and Dayan 1992) to modify the state transition rule and the trail update rule of the AS algorithm. The state transition rule is shown below:

$$v = \left\{ \arg \dots \max_{l \in u} \left[ (\tau_{il})^\alpha (\eta_{il})^\beta \right] \dots q \leq q_0 \right.$$

or

$$v = \{V \dots q > q_0\}$$

where  $q$  is a value chosen randomly with uniform distribution in  $(0,1)$ ,  $q_0$  is a parameter  $0 \leq q_0 \leq 1$ , and  $V$  is a random variable which is determined according to one of the following three rules:

- Pseudo-Random rule  
 $V$  is selected according to the uniform distribution.
- Pseudo-Random-Proportional rule

$V$  is selected according to the probability 
$$P_{iv} = \frac{(\tau_{iv})^\alpha (\eta_{iv})^\beta}{\sum_{l \in u} (\tau_{il})^\alpha (\eta_{il})^\beta}$$

- Random-Proportional rule  
 $q_0$  is set to zero, i.e., all decisions are based on the probability distribution

$$P_{iv} = \frac{(\tau_{iv})^\alpha (\eta_{iv})^\beta}{\sum_{l \in u} (\tau_{il})^\alpha (\eta_{il})^\beta}$$

Gambardella *et al.* indicate that the pseudo-random-proportional rule is superior to the other two rules. After all ants complete the construction of a tour, the pheromone trail is updated by

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} + (1 - \rho) \left[ \Delta \tau_{ij}^e + \gamma \max \tau_{jz} \right]$$

where  $\Delta \tau_{ij}^e$  uses the best solution in the iteration

or the best solution found so far, and  $\gamma \max \tau_{jz}$  accounts for the maximum pheromone trail of the next state multiplied by a discount factor.

### 3.5 Ant Colony System (ACS):

Dorigo and Gambardella propose the ACS algorithm, which is adapted from the Ant-Q algorithm. In order to balance the exploitation of good solutions and the exploration of search space, the pseudo-random-proportional rule shown as follows is used for the solution construction process.

$$v = \left\{ \arg \dots \max_{l \in u} \left[ (\tau_{il})(\eta_{il})^\beta \right] \dots q \leq q_0 \right.$$

or

$$v = \{V \dots q > q_0\}$$

and  $V$  is selected according to the probability 
$$P_{iv} = \frac{(\tau_{iv})(\eta_{iv})^\beta}{\sum_{l \in u} (\tau_{il})(\eta_{il})^\beta}$$
 The  $\alpha$  is set to 1 in ACS because

Gambardella and Dorigo (1995) show that it gives the best result. Thereafter, the pheromone update rule consists of two phases - Local Updating (online updating) and Global Updating (offline updating). The purpose of local updating is to decay the pheromone intensity of the selected move to give more chance to exploration.

Local updating is applied each time after an ant makes a move by  $\tau_{ij}^{new} = \rho\tau_{ij}^{old} + (1 - \rho)\Delta\tau_{ij}$  where  $\Delta\tau_{ij}$  can be  $\gamma \max \tau_{jz}$ ,  $\tau_0$  or zero. Dorigo *et al.* find that the former two provide similar performance and outperform the last one. Global updating is only applied after all ants have constructed their solutions by  $\tau_{ij}^{new} = \rho\tau_{ij}^{old} + (1 - \rho)\Delta\tau_{ij}^e$  where  $\Delta\tau_{ij}^e$  considers only the best solution found so far.

**3.6 Max-Min Ant System (MMAS):**

In order to avoid the stagnation situation in which all ants are stuck within a local optimum, StUzle and Hoos (Stiitzle and Hoos 1997, 1998, 1999, 2000) propose the MMAS algorithm to have more control on the pheromone trail. The state transition rule used is either the random-proportional rule or the pseudo-random-proportional rule. The pheromone trail is updated when all ants complete their solution construction by

$$\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \Delta\tau_{ij}^e$$

where either the best solution in this iteration and the best solution found so far

is used for  $\Delta\tau_{ij}^e$ . All  $\tau_{ij}$  are initialized as  $\tau_{max}$  and  $\tau_{min} \leq \tau_{ij} \leq \tau_{max}$ . StUzle and Dorigo (1999)

also propose a variation of the state transition rule as  $P_{ij} = \frac{\tau_{ij}}{\sum_{l \in u} \tau_{il}}$  because (as shown in an MMAS

application to the TSP) when local search is used to improve the algorithm, the importance of local heuristic information is replaced by local search. Therefore, local heuristic information is ignored in this version of state transition rule (Stiitzle and Dorigo 1999, Stutzle and Hoos 1997, 1998,1999,2000).

**3.7 ANTS:**

Mainezzo proposes an improved ACO algorithm for the QAP.  $\eta_{ij}$  is estimated by means of lower bounds, and the state transition rule is modified to

$$P_{ij} = \frac{\alpha.\tau_{ij} + (1 - \alpha).\eta_{ij}}{\sum_{l \in u} [\alpha.\tau_{il} + (1 - \alpha).\eta_{il}]}$$

Pheromone update does not use pheromone evaporation ,i.e.  $\tau_{il}^{new} = \tau_{ij}^{old} + \sum_{k=1}^{NA} \Delta\tau_{ij}^k$  where

$$\Delta\tau_{ij}^k = \tau_0 \cdot \left(1 - \frac{L^k - LB}{L_{avg} - LB}\right) \cdot L^k$$

is the  $K^{th}$  ant ,  $L_{avg}$  represent the moving average of the last W

ants, and LB denotes the value of the lower bound, which is calculated in the beginning of the algorithm.

Merkle and Middendorf (2000) propose an alterative method for pheromone evaluation where the pheromone values corresponding to older decisions are taken into account. Thus, the transition probability

$$P_{iv} = \frac{\left(\sum_{k=1}^i \tau_{kv}\right)^\alpha (\eta_{iv})^\beta}{\sum_{l \in u} \left(\sum_{k=1}^i \tau_{kl}\right)^\alpha (\eta_{il})^\beta}$$

After the initial proof-of-concept application to the traveling salesman problem (TSP) (Dorigo *et al.*, 1991,1996), ACO was applied to many other CO problems.

The first convergence proofs were presented by Gutjahr in (Gutjahr.2000, 2002). He proved convergence with probability 1 to the optimal solution (in (Gutjahr.2000)), and more in general to any optimal solution (in (Gutjahr.2002)), of a particular ACO algorithm that he called graph-based ant system (GBAS). Not withstanding its theoretical interest, the main limitation of this work was that GBAS is quite different from any implemented ACO algorithm and its empirical performance is unknown.

A second strand of work on convergence focused therefore on a class of ACO algorithms that are among the best-performing in practice, namely, algorithms that apply a positive lower bound  $\tau_{min}$  to all pheromone values. The lower bound prevents the probability to generate any solution to become zero. This class of algorithms is denoted by ACO min . Dorigo and Stutzle, first in (Stutzle, *et al.* 2002) and later in (Dorigo, *et al.* 1996), presented a proof for the convergence in value, as well as a proof for the convergence in solution, for algorithms from ACO\_min.

Recently, researchers have been dealing with the relation of ACO algorithms to other methods for learning and optimization. One example is the work presented in (Birattari, *et al.* 2002) that relates ACO to the fields of optimal control and reinforcement learning. A more prominent example is the work that aimed at finding similarities between ACO algorithms and other probabilistic learning algorithms such as stochastic gradient ascent (SGA), and the cross-entropy (CE) method.

Meuleau and Dorigo have shown in (Meuleau *et al.*, 2002) that the pheromone update as outlined in the proof-of-concept application to the TSP (Dorigo *et al.*, 1991, 1996) is very similar to a stochastic gradient ascent in the space of pheromone values.

Blum (2004) was proposed the first implementation of SGA-based ACO algorithms where it was shown that SGA-based pheromone updates avoid certain types of search bias. Zlochin *et al.* (Zlochin *et al.*, 2004) have proposed a unifying framework for so-called model-based search (MBS) algorithms. An MBS algorithm is characterized by the use of a (parametrized) probabilistic model. The class of MBS algorithms can be divided into two subclasses with respect to the way the probabilistic model is used. The algorithms in the first subclass use a given probabilistic model without changing the model structure at run-time, whereas the algorithms of the second subclass use and change the probabilistic model in alternating phases.

ACO algorithms are examples of algorithms from the first subclass. While convergence proofs can provide insight into the working of an algorithm, they are usually not very useful to the practitioner that wants to implement efficient algorithms. This is because, generally, either infinite time or infinite space are required for a stochastic optimization algorithm to converge to an optimal solution (or to the optimal solution value). The existing convergence proofs for particular ACO algorithms are no exception.

In (Blum *et al.*, 2005, 2004), Blum and Dorigo adopted the term deception for the field of ant colony optimization, similarly to what had previously been done in evolutionary computation. It was shown that ant colony optimization algorithms in general suffer from first order deception in the same way as GAs suffer from deception. Blum and Dorigo further introduced the concept of second order deception, which is caused by a bias that leads to decreasing algorithm performance over time.

Recently, Montgomery *et al.* (Montgomery *et al.*, 2004) made an attempt to extend the work by Blum and Sampels (Blum, *et al.* , 2002) to assignment problems, and to attribute search bias to different algorithmic components.

Merkle and Middendorf (2002) were the first to study the behavior of a simple ACO algorithm by analyzing the dynamics of its model, which is obtained by applying the expected pheromone update. Their work deals with the application of ACO to idealized permutation problems. When applied to constrained problems such as permutation problems, the solution construction process of ACO algorithms consists of a sequence of random decisions in which later decisions depend on earlier ones. Therefore, the later decisions of the construction process are inherently biased by the earlier ones. The work of Merkle and Middendorf shows that this leads to a bias which they call selection bias. Furthermore, the competition between the ants was identified as the main driving force of the algorithm.

Deneubourg *et al* (1990) tested ants' foraging behavior by a single bridge experiment with each branch has the same length. Initially, there is no pheromone on the either branch, and the first few ants have the same probability to randomly select either branch. A few more ants may select one branch, say upper branch, over the other. More ants mean a larger amount of pheromone is deposited to the upper branch, which in turn invites more ants to visit the upper branch. In this a single factor experiment, the behavior of the ants was found to be controlled by pheromone.

Goss *et al* (1989) considered the double bridge experiment, in which the longer branch is twice as long as the short branch. The individual ants travel from the nest to the food source and return to the nest by the same pheromone driven mechanism as in the single bridge situation. Those ants that took the shortest path forward and back will return to the nest first, and immediately after these ants return, more pheromone is left on shorter branch than on the longer branch. By the time after ants arrive, pheromone left by the ants that selected the longer branches will have evaporated soon than in the shorter branches. As more ants deposit more pheromone to the shorter branches, and more pheromones attract more ants, the shortest path is identified. As the length ratio between the two branches increases, more ants will select shorter branch (Bonabeau, 1996).

Botee and Bonabeau (1998) further studied the selection of parameters with genetic algorithms. Wu *et al.* (1999) put forward an ACA with mutation features by introducing mutation into the fundamental ACA and utilizing the conciseness and effectiveness of the 2-exchange method. The principles of the algorithm can be illustrated by examining the food searching process of an ant colony.

Krzysztof Walkowiak (2004) Their research was about algorithms to the non-bifurcated multicommodity flow problem. They proposed a general framework of ant algorithm that can be applied to the design of static flows in connection-oriented computer networks. Next, through numerical simulation, they studied the influence of algorithm's parameters setting on the quality of solutions. They compare and discuss two variants of the algorithm: without and with feasible initial solution.

Abbaspour *et al.* (2001) used the ACO algorithm for estimating the unsaturated soil hydraulic parameters. Zecchin *et al.* (2003) compared the performance of original ant system with that of Max–Min Ant System (MMAS), a modified version of the ant system proposed by Stutzle and Hoos (2001), for optimization of water distribution networks.

Simpson *et al.* (2001) discussed the selection of parameters used in the ACO algorithm for pipe network optimization problems.

Maier *et al.* (2003) compared the performance of the ACO algorithm with that of GAs for the optimization of water distribution networks.

Afshar (2005) proposed a new transition rule for ACO algorithms using elitist strategies and applied the method to pipe network optimization problems. That method was shown to overcome the premature convergence problem encountered by elitist ACO algorithms while improving the convergence characteristics of the algorithms compared to alternative methods such as MMAS.

## REFERENCES

Abbaspour, K.C., R. Schulin, M.Th. van Genachten, 2001. Estimating unsaturated soil hydraulic parameters using ant colony optimization. *Advances in water Resources*, 24: 827-841.

Afshar, M.H., 2005. A new transition rule for ant colony optimisation algorithms: application to pipe network optimisation problems. *Eng Optim*, 37(5): 525–40.

Aytak, A., O. Kisi, 2008. A genetic programming approach to suspended sediment modelling. *Journal of Hydrology*, 351: 288–298.

Birattari, M., G. Di Caro, M. Dorigo, 2002. Toward the formal foundation of ant programming, in: M. Dorigo, G. Di Caro, M. Sampels (Eds.), *Ant Algorithms*, Proc. ANTS, Third Internat. Workshop, Lecture Notes in Computer Science, Springer, Berlin, Germany, 2463: 188–201.

Blum, C., 2004. *Theoretical and Practical Aspects of Ant Colony Optimization*, Dissertations in Artificial Intelligence, 282, Akademische Verlagsgesellschaft Aka GmbH, Berlin, Germany.

Blum, C., M. Dorigo, 2004. Deception in ant colony optimization, in: M. Dorigo, M. Birattari, C. Blum, L.M. Gambardella, F. Mondada, T. Stutzle (Eds.), Proc. ANTS 2004, Fourth Internat. Workshop on Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Computer Science, Springer, Berlin, Germany, 317: 119–130.

Blum, C., 2005. Beam-ACO- Hybridizing ant colony optimization with beam search: an application to open shop scheduling, *Comput. Oper. Res.*, 32(6): 1565-1591.

- Blum, C., M. Dorigo, 2005. Search bias in ant colony optimization: on the role of competition-balanced systems, *IEEE Trans. Evol. Comput.*, 9(2): 159–174.
- Blum, C., A.I. Rol, 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Comput. Surveys*, 35(3): 268–308.
- Blum, C., M. Sampels, 2002. Ant Colony Optimization for FOP shop scheduling: a case study on different pheromone representations, *Proc. Congr. on Evolutionary Computation (CEC'02)*, Vol. 2, IEEE Computer Society Press, Los Alamitos, CA: 1558–1563.
- Blum, C., M. Sampels, 2002. When model bias is stronger than selection pressure, in: J.J. Merelo Guervos *et al.*(Eds.), *Proc. PPSN-VII, Seventh Internat. Conf. on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, Vol. 2439, Springer, Berlin, Germany: 893–902.
- Blum, C., M. Sampels, 2004. An ant colony optimization algorithm for shop scheduling problems, *J. Math. Model. Algorithms*, 3(3): 285–308.
- Bonabeau, E., 1996. Marginally stable swarms are flexible and efficient. *J. Phys. I France*, 6: 309–320.
- Botee, H.M., E. Bonabeau, 1998. Evolving ant colony optimization, *Complex Systems*, 1(2): 149–59.
- Bullnheimer, B., R.F. Hartl, C. Strauss, 1999. "A New Rank Based Version of the Ant System - A Computational Study," *Central European Journal for Operations Research and Economics*, 7: 25-38.
- Colomi, A., M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini and M. Trubian, 1996. "Heuristics from Nature for Hard Combinatorial Optimization Problems," *International Transactions in Operational Research*, 3(1): 1-21.
- Deneubourg, J.L., S. Goss, 1989. Collective Patterns and Decision Making, *Ethology, Ecology and Evolution I*: 295-311.
- Deneubourg, J.L., S. Aron, S. Goss and J.M. Pasteels, 1990. The self-organizing exploratory pattern of the argentine ant, *J. Insect Behav.*, 3: 159–168.
- Dorigo, M., 1992. Optimization, learning and natural algorithms (in Italian), Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.
- Dorigo, M., V. Maniezzo, A. Colomi, 1991. Positive feedback as a search strategy, *Tech. Report 91-016*, Dipartimento di Elettronica, Politecnico di Milano, Italy.
- Dorigo, M., V. Maniezzo, A. Colomi, 1996. Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Systems, Man, Cybernet.-Part B* 26(1): 29–41.
- Dorigo, M., L.M. Gambardella, 1997. Ant colonies for the travelling salesman problem. *BioSystems*, 43: 73–81.
- Dorigo, M., T. Stutzle, 2004. *Ant Colony Optimization*, MIT Press, Cambridge, MA. Fogel, L.J., Owens, A.J., Walsh, M.J. 1996. *Artificial Intelligence Through Simulated Evolution*, Wiley, New York.
- Gambardella, L.M., E.D. Taillard and G. Agazzi, 1999. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows, in: D. Corne, M. Dorigo, F. Glover, F., 1989. Tabu search—Part I, *ORSAJ. Comput.*, 1(3): 190–206.
- Glover, F., 1990. Tabu search—Part II, *ORSAJ. Comput.*, 2(1): 4–32.
- Glover, F., Kochenberger, G(Eds.), 2002. *Handbook of Metaheuristics*, Kluwer Academic Publishers, Norwell, MA.
- Glover, F., M. Laguna, 1997. *Tabu Search*, Kluwer Academic Publishers, Dordrecht
- Goss, S., S. Aron, J.L. Deneubourg and J.M. Pasteels, 1989. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76: 576–581.
- Gutjahr, W.J., 2000. A graph-based ant system and its convergence, *Future Gen. Comput. Systems*, 16(9): 873–888.
- Gutjahr, W.J., 2002. ACO algorithms with guaranteed convergence to the optimal solution, *Inform. Process. Lett.*, 82(3): 145–153.
- Holland, J.H., 1975. *Adaption in Natural and Artificial Systems*, The University of Michigan Press, Ann Harbor, MI.
- Hoos, H.H, T. Stutzle, 2004. *Stochastic Local Search: Foundations and Applications*, Elsevier, Amsterdam, The Netherlands.
- Kaji, T., 1999. "Parallel Revised Ant System Algorithm by New Agents for TSP," *Proceedings of the Second Asia-Pacific Conference on Industrial Engineering and Management System (APIEMS'99)*, Kanzawa, Japan, October: 639-642.
- Kirkpatrick, S., C.D. Gelatt, M.P. Vecchi, 1983. Optimization by simulated annealing, *Science*, 220(4598): 671–680.
- Krzysztof, 2004. *LNAI(3070)*, pp: 922-927.

Maniezzo, V., 1996. "Exact and Approximate Nondeterministic Tree-Search Procedures for the Quadratic Assignment Problem," Research Report CSR 98-1, 1998, Università di Bologna, Italy Marshall, T.J., J.W. Holmes, and C.W. Rose. Soil Physics (Third Edition). 453 pp. New York, Cambridge University Press.

Maier, H.R., A.R. Simpson, A.C. Zecchin, W.K. Foong, K.Y. Phang, H.Y. Seah, *et al.*, 2003. Ant colony optimization for design of water distribution systems. *J Water Resour Plan Manag*, ASCE, 129(3): 200–209.

Merkle, D., M. Middendorf and H. Schmeck, 2002. Ant colony optimization for resource-constrained project scheduling, *IEEE Trans. Evol. Comput.*, 6(4): 333–346.

Meuleau, N., M. Dorigo, 2002. Ant colony optimization and stochastic gradient descent, *Artif. Life*, 8(2): 103–121.

Merkle, D., M. Middendorf, 2002. Modelling ACO: composed permutation problems, in: M. Dorigo, G. Di Caro, M. Sampels (Eds.), *Ant Algorithms*, Proc. ANTS 2002, Third Internat. Workshop, Lecture Notes in Computer Science, Vol. 2463, Springer, Berlin, Germany, pp: 149–162.

Merkle, D., M. Middendorf, 2002. Modelling the dynamics of ant colony optimization algorithms, *Evol. Comput.*, 10(3): 235–262.

Sichingabula, H.M., 1993. Character and controls of suspended-sediment concentration and discharge effectiveness Fraser River, British Columbia, Canada, Simon Fraser University.

Stutzle, T., M. Dorigo, 2002. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, Kluwer Academic Publishers, Norwell, MA.

Stutzle, T., H. Hoos, 1997. The Max-Min Ant System and local search for the traveling salesman problem. In T. Bick, Z. Michalewicz, and X. Yao, editors, *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, pages 3093–314. IEEE Press, Piscataway, NJ.

Stutzle, T., H. Hoos, 1999. Max-min ant system and local search for combinatorial optimization problems, in S. Voß, S. Martello, I. H. Osman, and C. Roucairol (eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, MA, pp. 313–29. Boston: Kluwer Academic.

Stutzle, T., H.H. Hoos, 2000. MAX-MIN ant system, *Future Gen. Comput. Systems*, 16(8): 889–914.

Zecchin, A.C., H.R. Maier, A.R. Simpson, A. Roberts, M.J. Berrisford, 2003. Leonard M. Max–Min ant system applied to water distribution system optimization. *Modsim 2003 – international congress on modelling and simulation*, 14–17 July, vol. 2. Townsville (Australia): Modelling and Simulation Society of Australia and New Zealand Inc., 2: 795–800.

Zlochin, M., M. Birattari, N. Meuleau and M. Dorigo, 2004. Model-based search for combinatorial optimization: a critical survey, *Ann. Oper. Res.*, 131(1–4): 373–395.