Ant Colony Optimization

ACO has a repetitive structure, which updates the pheromone trails in each iteration with regards to the direction and usage frequency of each path followed by artificial ants to move on the search graph, representing the problem to solve.

From: Decision-Making for Biomass-Based Production Chains, 2019

Related terms:

Energy Engineering, Particle Swarm Optimization, Swarm Intelligence, Genetic Algorithm, Optimisation Problem, Search Algorithm

View all Topics

Energy optimization in low-power wide area networks by using heuristic techniques

Zeinab E. Ahmed, ... Sheetal N. Ghorpade, in LPWAN Technologies for IoT and M2M Applications, 2020

10.5.3 Ant colony optimization algorithm

ACO is inspired by the foraging behavior of ants [31]. At the core of this behavior is the indirect communication between the ants with the help of chemical pheromone trails, which enables them to find short paths between their nest and food sources. Blum [32] exploited this characteristic of real ant colonies in ACO algorithms to solve global optimization problems. Dorigo [33] developed the first ACO algorithm and since then numerous improvements of the ant system have been proposed. ACO algorithm has strong robustness as well as good dispersed calculative mechanism. ACO can be combined easily with other methods; it shows well performance in resolving the complex optimization problem. ACO optimizes a problem by having an updated pheromone trail and moving these ants around in the search space according to simple mathematical formulae over the transition probability and total pheromone in the region. At each iteration, ACO generates global ants and calculates their fitness. Update pheromone and edge of weak regions. If fitness is improved, then move local ants to better regions, otherwise select new random search direction. Update ant pheromone and evaporate ant pheromone. The continuous ACO is based on both local and global search. Local ants have the capability to move toward latent region with best solution with respect to transition probability of region *k*,

(10.3)

where total pheromone at region *k* and *n* is number of global ants.

Pheromone is updated using the following equation:

(10.4)

where *r* is the pheromone evaporation rate.

The probability of selection of region for local ants is proportional to pheromone trail. The process flow for ACO is shown in Fig. 10–4.

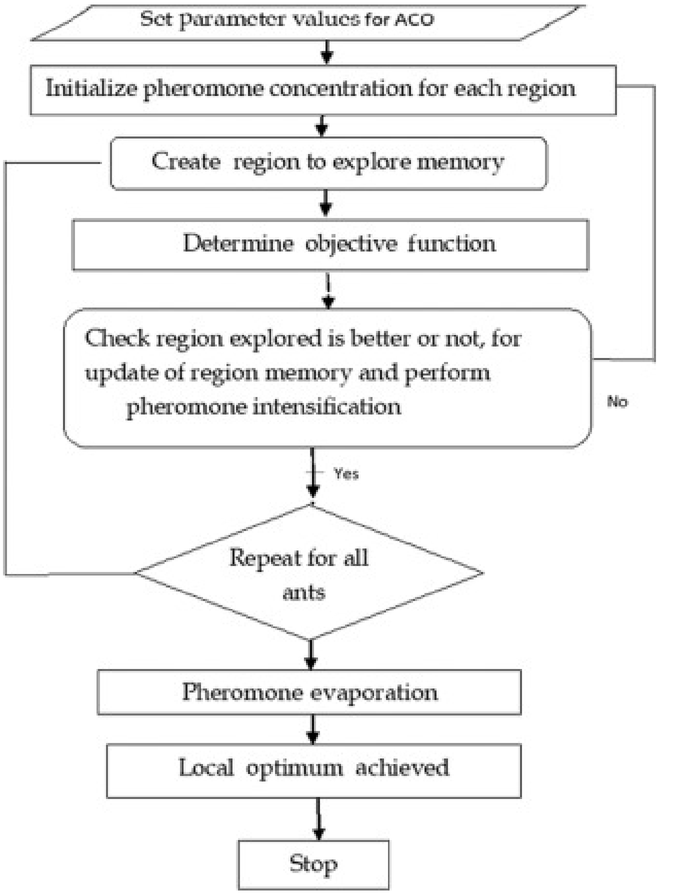
Set parameter values for ACO
Initialize pheromone concentration for each region
Create region to explore memory
Determine objective function
Check region explored is better or not, for update of region memory and perform pheromone intensification
No
Yes
Repeat for all ants
Pheromone evaporation
Local optimum achieved
Stop

Figure 10–4. Process flow of ant colony optimization.

> Read full chapter

Nature-inspired computation and swarm intelligence: a state-of-the-art overview

Xin-She Yang, Mehmet Karamanoglu, in Nature-Inspired Computation and Swarm Intelligence, 2020

1.3.2 Ant colony optimization

ACO, developed by Marco Dorigo in 1992 (Dorigo, 1992), was the first swarm intelligence-based algorithm. In essence, ACO mimics the foraging behavior of social ants in a colony, and pheromone is used for simulating the local interactions and communications among ants. Pheromone is deposited by each ant and it evaporates gradually with time. The exact form of evaporation model may vary, depending on the variant and form of ACO used in implementations. Both incremental deposition and exponential decay of pheromone are widely used.

ACO is particularly suitable for discrete optimization problems. For example, for routing problems, a route or path is encoded as a solution. When ants explore different paths, explored routes are marked with deposited pheromone that evaporates over time. The fitness or quality of a path (a solution) is related to the concentration of pheromone on the path. Routes with higher pheromone concentrations will be preferred or be chosen with a higher probability at a junction. Similar to GA, ACO is a mixed procedure with many variants and applications (Dorigo, 1992).

> Read full chapter

Optimization of Methods for Image-Texture Segmentation Using Ant Colony Optimization

Manasa Nadipally, in Intelligent Data Analysis for Biomedical Applications, 2019

2.5.5 Comparison of Results

ACO metaheuristics are specialized in solving the optimization problems in every iteration step by vaporizing and updating the pheromone density. In Figs. 2.4–2.6 ACO image segmentation with a different number of iterations using different techniques are illustrated. Fig. 2.4 depicts the brain MRI images segmented with 100 and 200 iterations using the ACO technique. Increasing the number of iterations improves the segmentation performance of ACO using a lower number of ants, as depicted in Fig. 2.4. The decrease in some ants to 1% of the total number of pixels and the increase in the number of iterations from 20 to 200 results in significantly improved segmented images with a higher probability of edge detection.

The comparison of ACO performance with conventional edge detection techniques is shown in Figs. 2.5 and 2.6, respectively. Fig. 2.5 shows the comparison of brain MRI image segmentation using algorithms of Canny, Artificial Bee, Sobel, Robert, Prewitt, and ACO techniques. The traditional edge detection techniques show unclear edges with poor segmentation. However, the results of the ACO technique depict the compensation of the results regarding edge detection for suitable image segmentation [17].

The ACO method utilized two thresholds to identify weak and strong edges. It only incorporated those weak edges in the output which were linked to strong edges. Consequently, the technique is accurate and robust to noise with higher probability of detecting the true weak/strong edges.

Additionally, the proposed method exploits the movement of a definite number of ants on the image based on the localized disparity in the image's intensity value. The localized disparity in the image's intensity value is used to develop a pheromone matrix, which provides the edge data of the image.

The statistical comparison of the ACO technique with other traditional techniques is carried out using the operator parameters which include RMSE, SNR, and PSNR. Tables 2.3 and 2.4 show the comparative values of the operator parameters for five iris and five brain MRI images evaluated through six algorithms (Canny, Sobel, Robert, Artificial Bee, Prewitt, and ACO). RMSE provides the difference between the values of the edge-detected pixels and the original pixels for a particular technique. Low RMSE signifies less difference between the values of the original and processed images, which indicates the accuracy of the output image. SNR is a relative value of the signal concerning noise and must be high for the accuracy of the output. On the other hand, the PSNR peak should be higher for enhanced quality.

Table 2.3 shows the comparative results for RMSE, SNR, and PSNR values obtained by different detector types (algorithms) used on MRI images. Although the ACO algorithm offers more clear edges, a comparative study of different edge detection techniques was carried out for validating the boundary extraction of the images. Table 2.3 shows that RMSE for ACO is at a minimum (3.7519) which indicates higher accuracy. On the other hand, comparatively higher values of SNR and PSNR (82.3721 and 36.6483, respectively) depict higher image quality for ACO in MRI images.

Table 2.4 shows the comparative results for RMSE, SNR, and PSNR values obtained by different detector types (Canny, Robert, Sobel, Prewitt, ACO, etc., algorithms) used on iris images. Although the ACO algorithm offers more clear edges, a comparative study of the different edge detection techniques was carried out for validating the boundary extraction of the images. Table 2.4 shows that RMSE for ACO is at a minimum (1.9119) which indicates higher accuracy. On the other hand, comparatively higher values of SNR and PSNR (65.4980 and 42.5026, respectively) depict higher image quality for ACO in the iris images.

The following figures display the graphs showing the comparison of RMSE, SNR, and PSNR values for 10 (5 iris and 5 brain MRI) images.

Fig. 2.7 shows a graph for depicting the RMSE performance of different algorithms using five iris and five brain MRI images. It can be observed that ACO peaks are smallest among all, which indicates lowest RMSE. On the other hand, Robert and Sobel's detectors exhibit highest RMSE values, which indicate lower accuracy in image segmentation. Similarly, Figs. 2.8 and 2.9 show the SNR and PSNR for different algorithms applied on iris and brain MRI images. Both graphs display higher ACO peaks, which indicate higher segmentation and edge detection quality for the ACO-based technique. On the other hand, Robert and Prewitt's techniques show lowest peaks, which indicate their poor segmentation characteristics.

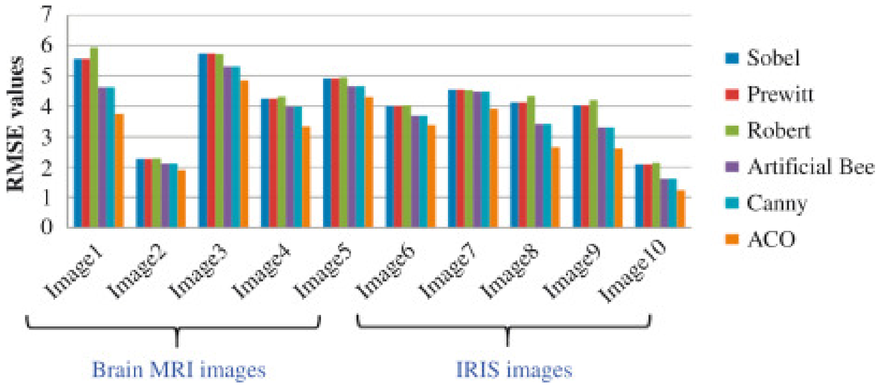

Figure 2.7. Comparison of image edge detection methods using root-mean-square-error (RMSE).

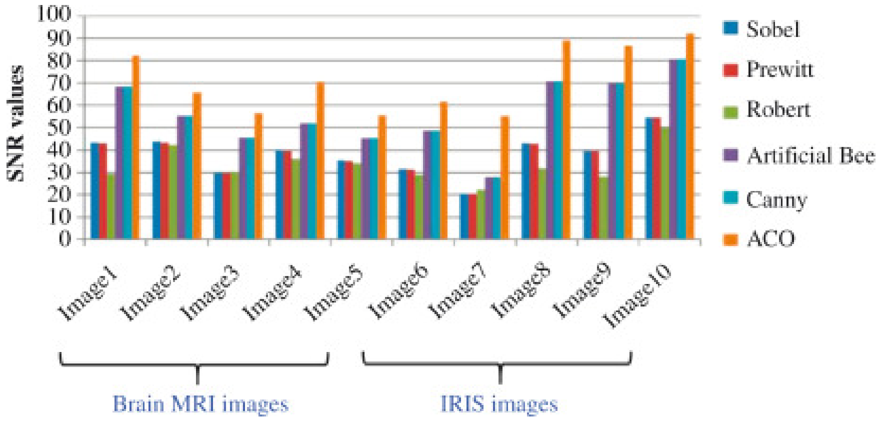

Figure 2.8. Comparison of image edge detection methods using signal-to-noise-ratio (SNR).

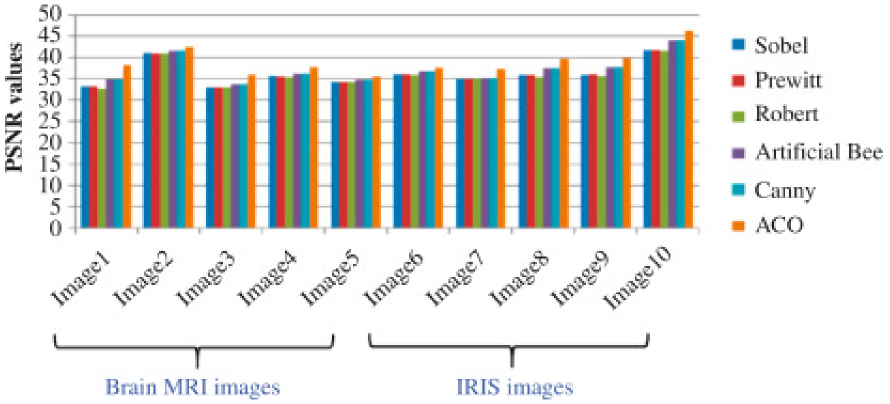

Figure 2.9. Comparison of image edge detection methods using peak-signal-to-noise ratio (PSNR).

An adaptive edge detector is essential for obtaining a robust solution which is adaptable to the fluctuating noise levels of the images and helps to distinguish valid contents of images from visual defects induced by noise. The ACO's performance heavily depends on the changeable parameters (e.g., □, typically known as the standard deviation of the Gaussian filter) and threshold values (e.g., T_1 and T_2). □ also regulates the Gaussian filter's size. Higher values of □ result in larger sizes of the Gaussian filter. This phenomenon implies higher and more blurring levels, essential for noisy images along with detection of the larger edges. The results and graphs shown in this chapter depict the higher effectiveness of the indigenously modified ACO algorithm proposed. The tuning of the parameters has yielded excellent results (better than the previous results quoted in the literature) concerning lower RMSE and higher SNR and PSNR, as indicated in Tables 2.3 and 2.4. Our main aim was to improve the image segmentation with enhanced image edge detection by lowering the error margin and enhancing the image quality. The task has been accomplished using the Java framework and modified ACO algorithm.

> Read full chapter

Optimum Performance-Based Seismic Design of Frames Using Metaheuristic Optimization Algorithms

Siamak Talatahari, in Metaheuristic Applications in Structures and Infrastructures, 2013

17.5.4 PSACO Algorithm

The PSACO, a hybridized approach based on PSO and ACO, is described in this section. PSACO utilizes a PSO algorithm as a global search, and the ACO approach worked as a local search. PSACO was utilized to solve different civil engineering examples such as a benchmark engineering problem (Kaveh and Talatahari, 2009b), truss design with continuous variables (Kaveh and Talatahari, 2009a), truss design with discrete variables (Kaveh and Talatahari, 2009c), frame optimization (Kaveh and Talatahari, 2008; Kaveh and Talatahari, 2009d), as well as many others.

Compared to other evolutionary algorithms based on heuristics, the advantages of PSO consist of easy implementation and a smaller number of parameters to be adjusted. However, it is known that the original PSO had difficulties in controlling the balance between exploration (global investigation of the search place) and exploitation (the fine search around a local optimum). In order to improve this character of PSO, one method is to hybridize PSO with other approaches such as ACO.

The implementation of PSACO algorithm consists of two stages. In the first stage, it applies PSO, while ACO is implemented in the second stage. ACO works as a local search, wherein, ants apply pheromone-guided mechanism to refine the positions found by particles in the PSO stage. In PSACO, a simple pheromone-guided mechanism of ACO is proposed to be applied for the local search. The proposed ACO algorithm handles *M* ants equal to the number of particles in PSO.

In the ACO stage, each ant generates a solution around which can be expressed as:

(17.25)

In the above equation, denotes a random number normally distributed with mean value and variance *i*, where:

(17.26)

 is used to control the step size. The normal distribution with mean can be considered as a continuous pheromone that has the maximum value in and decreases with going away from it. In ACO algorithms, the probability of selecting a path with more pheromone is greater than other paths. Similarly, in the normal distribution, the probability of selecting a solution in the neighborhood of is greater than the others. This principle is used in the PSACO algorithm as a helping factor to guide the exploration and to increase the controlling in exploitation.

In the present method, the objective function value of the new solution in the ACO stage is computed and if it is better, the current position of ant *i*, is replaced by the current position of particle *i* in the swarm.

> Read full chapter

Symbol Detection in Multiple Antenna Wireless Systems via Ant Colony Optimization

Manish Mandloi, Vimal Bhatia, in Handbook of Neural Computation, 2017

12.2 Overview of Ant Colony Optimization

ACO is a well known bio-inspired technique used for solving the combinatorial optimization problems. ACO was initially proposed by Marco Dorigo in his Ph.D. thesis in the early 90s [31] aiming to solve the optimal path problem in a graph. Inspired by the foraging behavior of natural ants where a colony of ants seeks the shortest path between the food source and their nest, the artificial ACO algorithm performs the similar process. In ACO algorithm, a colony of artificial ants is considered which mimics the foraging behavior of natural ants to find a solution as depicted in the flow chart in Fig. 12.1. Each ant starts its journey from the nest in search of food source and comes back to the nest which completes an iteration. During the journey each ant lay a substance known as *pheromone* on the path they travel. The concentration of pheromone on each path depends on the distance of the path and the quality of food source available. The amount of pheromones on a path plays a significant role in selection of the path, i.e. the more the pheromone concentration on a path, the more will be its chances of being selected by ants. Each ant probabilistically selects a path which depends on the pheromone concentration and some heuristic value such as the objective function value. The pseudo code of the basic ACO is shown in Algorithm 12.1.

Figure 12.1. Flow chart of the basic ACO algorithm.

Algorithm 12.1. Ant Colony Optimization

> Read full chapter

Large-Scale Maintenance Optimization Problems for Civil Infrastructure Systems

Sehyun Tak, ... Hwasoo Yeo, in Metaheuristic Applications in Structures and Infrastructures, 2013

22.3.2 Ant Colony Optimization

ACO, developed by Dorigo (Dorigo and Stützle, 2004), is an iterative algorithm and belongs to the class of metaheuristics. It is based on the foraging behavior of ants in nature, which are capable of finding the shortest path between their nest and a source of food by stigmergy, which is an indirect form of communication.

The ACO algorithm is composed of three parts: ant-based solution construction, pheromone update, and iteration (Blum, 2005). In the ant-based solution construction phase, the sequences of solution components represent the series of artificial ants' position changes that are determined by the stochastic-mechanism based pheromone. One solution component refers to an ant's position, which represents the repair action in the infrastructure maintenance management system. In the ACO algorithm, pheromone update is the important phase because the differences between ACO algorithms such as ant system (AS), max–min ant system (MMAS), and ant colony system (ACS) are caused by the difference of pheromone rules (Dorigo et al., 2006). In AS, pheromone values are updated by all ants that pass the node. In AS, even though a path is insufficient, if the number of ants that have passed through the path is great, too much pheromone can be accumulated on the insufficient path. In other words, due to the amount of excessively accumulated pheromone on the path, the insufficient path may be recognized and selected incorrectly as the good path. MMAS is an algorithm that tries to overcome this shortcoming of AS. Unlike AS, in which pheromone is updated time to time during the construction process, only the pheromone value of the best ant is updated at the end of the construction process in MMAS. ACS is similar to MMAS in terms of updating pheromone at the end of the process, but the difference is that in ACS, the processes of local pheromone updates are included during iterations to diversify the search.

ACO shows better performance than GA in terms of computation time when optimizing maintenance dates because, in such cases, a priori information can be included in the information matrix (Samrout et al., 2005). Due to the strength of this algorithm, ACO has been applied to several optimization problems such as scheduling problems, vehicle routing problems, the set packing problem, transportation networks, and preventive maintenance (Blum, 2005; Gandibleux et al., 2005; Merkle et al., 2002; Samrout et al., 2005; Vitins and Axhausen, 2009). Generally, ACO algorithms are competitive with other optimization techniques when applied to the problems that are not overly constrained. However, ACO has limitations in determining the optimal repair action in situations where many constraints exist and there is less a priori information about the problem. Especially for the infrastructure maintenance management system, ACO has some drawbacks related to accuracy and processing time.

> Read full chapter

A Review on Traditional and Modern Structural Optimization

Mohammed Ghasem Sahab, ... Amir Hossein Gandomi, in Metaheuristic Applications in Structures and Infrastructures, 2013

2.5.4 Ant Colony Optimization

ACO was developed by Dorigo and his associates in the early 1990s (Colorni et al., 1991, Dorigo, 1992, Dorigo et al., 1996). The main idea of ACO is to imitate the cooperative behavior of an ant colony, which finds the shortest path to a food source. In this method, a combinatorial optimization problem with n design variables (x_1–x_n) is modeled as a multilayered graph as shown in Figure 2.1. The number of layers is equal to the number of design variables and the number of nodes in each layer is equal to the number of discrete values permitted for the corresponding design variable. Thus, each node in a specific position of the graph is associated with a permissible discrete value of a design variable. Artificial ants walk through this graph, looking for good paths. An ant colony consists of N ants. The ants start at the nest node, travel through the various layers from the first layer to the last layer, and end at the destination node in each cycle or iteration. Each ant can select only one node in each layer in accordance with the state transition rule given by metaheuristic information. The nodes selected along the path visited by an ant represent a candidate solution. A typical path visited by an ant is shown by thick lines in Figure 2.4. Once the path is complete, the ant deposits some pheromone on the path based on a local updating rule. In the beginning of the optimization process (i.e., in iteration 1), all the edges or rays are initialized with an equal amount of pheromone. As such, in iteration 1, all the ants start from the home node and end at the food node by randomly selecting a node in each layer. Small quantities of pheromone are deposited during the construction phase, while larger amounts are deposited at the end of each iteration in proportion to solution quality. The optimization process is terminated if any of the defined termination conditions are satisfied. The values of the design variables denoted by the nodes on the path with the largest amount of pheromone are considered as the components of the optimum solution vector. In general, at the optimum solution, all ants travel along the same best (converged) path.

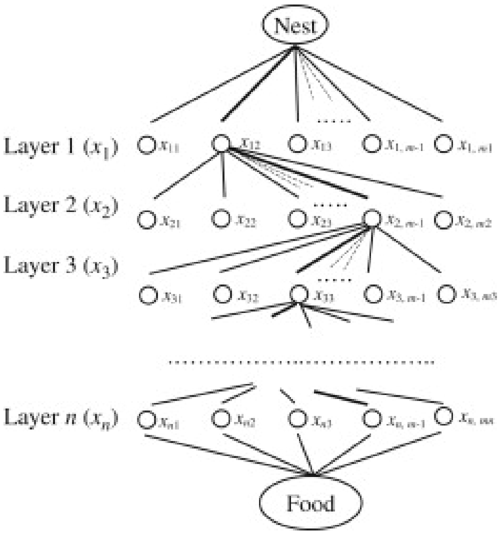

Figure 2.4. The ACO process in the form of a multilayered graph.

One of the first applications of ACO in structural engineering was presented by Bland (2001) for design optimization of a 25-bar space truss. Camp and Bichon (2004) also employed ACO for design optimization of space trusses. Later, they extended this work to optimize rigid steel frames (Camp et al., 2004). Kaveh and Shojaee (2007) employed ACO for optimal design of skeletal structures. A hybridization of ACO and other metaheuristic techniques, like PSO, have been applied for design optimization of steel frames and truss structures (Kaveh and Talatahari, 2007; Kaveh and Talatahari, 2009). ACO has been employed to determine the optimum design of 3D irregular steel frames, taking into account warping deformations of thin-walled sections (Aydoğdu and Saka, 2009). ACO has been used for structural topology optimization (Luh and Lin, 2009). Zhang and Li (2011) presented a two-level optimization algorithm based on ACO to design the shape of a transmission tower. Majumdar et al. (2012) utilized ACO to identify damages in truss structures.

> Read full chapter

Metaheuristic Applications in Highway and Rail Infrastructure Planning and Design

Manoj K. Jha, in Metaheuristics in Water, Geotechnical and Transport Engineering, 2013

16.4.2 Ant Colony Optimization

ACO is inspired by the concept of self-organization of swarms and is derived from swarm intelligence. The fundamental idea is that ants organize themselves to travel to the food source and have the ability to follow each other. The two important properties of ACO that basically simulate the real ant system are as follows (Bonabeau et al., 1999):

- Stigmergy: This is a property that plays an important role in developing a collective behavior of the social insects. The stimulatory factor pheromone trail is secreted from an ant, the amount of which decides the preference for the next ant to choose a path. This basically depicts the property called *self-organization*.
- *Autocatalysis*: According to this property, the shorter the path, the sooner the pheromone is deposited by the ants, and the more ants use the shorter path. This ensures the fact that the algorithm introduces the chance of *rapid convergence* while heading toward the optimal solution. The important property of this algorithm is the decaying of pheromone, which influences the convergence by governing the amount of accumulation of pheromone in the paths. It ensures that the search process does not get stuck in the local optima.

The principle of ACO can be explained by the following simple example (Figure 16.1). Suppose that there are two different reversible paths, and available to a group of ants. The ants can travel in either direction, with the objective of deriving the shortest path. The leg is twice as long as the leg. The underlying concept is that ants lay pheromone along the traveled path, which evaporates over time. Thus, the shorter a traveled leg, the longer the pheromone lasts. An ant traveling in the direction is faced with two options, or at point C. The decision of choosing a path over the other at point C is purely arbitrary and has equal probability. But the probability of choosing the shorter path grows in time for the follower ants, as the pheromone trail left by preceding ants lasts longer on the shorter path. After sufficient time intervals, all ants converge to the shortest path. A three-time-interval scenario with a 30-ant sample is shown in Figure 16.1.

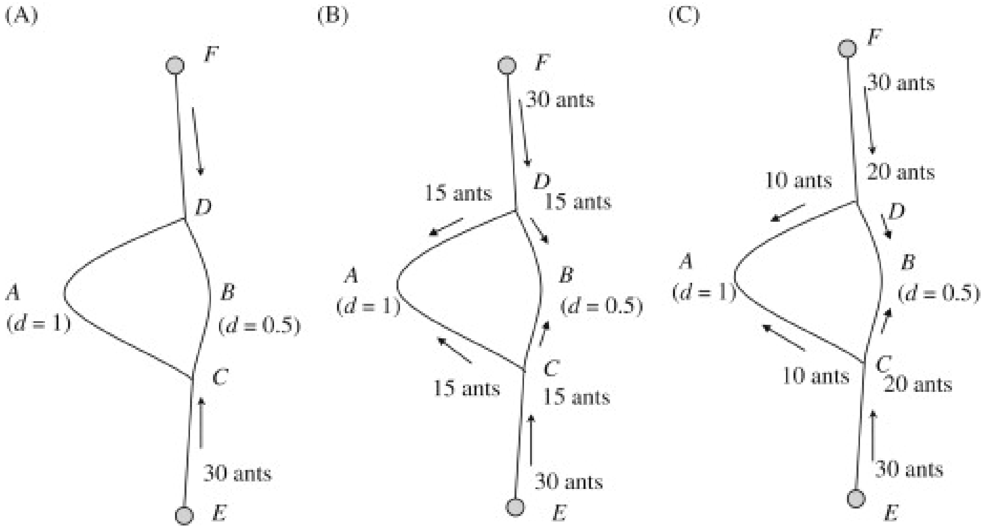

Figure 16.1. An example of ACO: (A) t=0, (B) t=1, and (C) t=2.

So far, ACO has been widely and successfully implemented for solving discrete optimization problems. It has been tried on both static and dynamic combinatorial optimization problems (Dorigo et al., 1999). A few examples of static optimization problem are given as follows:

Traveling salesman problem: In this problem, *n* cities are traveled in such a way that the total traveling cost is minimized. ACO has shown better performance than the GA for a small problem (30-city problem), but not for a larger problem (Dorigo et al., 1997).
Quadratic assignment problem: It is the problem of assigning *n* facilities in *n* locations so that the total cost of assignment is minimized. The results obtained for this problem is shown in Dorigo et al. (1999).
Job-shop scheduling problem: A set *J* of jobs are to be assigned on *M* machines in such a fashion that the total completion time is minimal, satisfying the constraint that no two jobs can be processed on the same machine at the same time. ACO was able to find 10% of the optimal value of the results for a 15-machine, 15-job problem (Dorigo et al., 1999).
*Vehicle routing problem***:** This problem is about obtaining minimum-cost vehicle routes for a fleet of vehicles starting from a depot or more than one depot. Dorigo et al. (1999) applied ACO on this problem and obtained reasonable results.

Some of the dynamic optimization problems where ACO is applied are the connection-oriented network routing problem and the connectionless network routing problem.

ACO for Searching in a Continuous Space

Although ACO has been applied to many discrete optimization problems, not many applications (Bilchev and Parmee, 1995; Kaveh and Talatahari, 2010) to continuous optimization problems have been observed. Bilchev and Parmee (1995) tried to solve a flight trajectory for an air-launched winged rocket that will achieve orbit before returning to atmosphere for a conventional landing by discretizing the continuous search space.

This approach says that the continuous nest neighborhood is divided in a finite number of directions represented by vectors (Figure 16.2). Feasible regions are first randomly placed in the search space, or they may correspond to regularly sampled directions from the nest. The agent then chooses a random direction and moves a short distance from the region's center in that direction with a probability proportional to the virtual pheromone concentration of the path that goes from the nest to the region. Agents reinforce their paths according to their performance, depending on the diffusion, evaporation, and recombination of the trails.

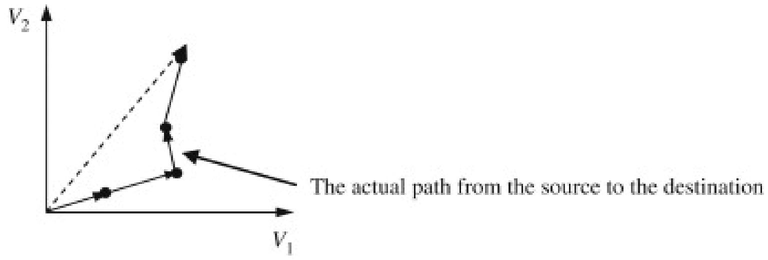

Figure 16.2. Discretization of a continuous search space.

Limitations of ACO

Essentially, then, it can be concluded that ACO fails to exploit a continuous search space the way that GA does. The continuous space must be discretized for ACO application, one of the inherent impediments in ACO application that is often ignored by researchers. Using the discretization concept for ACO application described in Bilchev and Parmee (1995), many real-world problems can be solved and results compared with those obtained with GAs. We have performed a comparative assessment of GA and ACO in highway and rail-alignment optimization in some of our recent works (Jha and Samanta 2006; Samanta and Jha, 2012). Using the discretization concept, we attempted to reformulate the highway alignment optimization (HAO) problem in order to seek its solution with ACO. Preliminary analyses were presented in some of our earlier works (Jha, 2001, 2002).

> Read full chapter

Heuristic and Meta-Heuristic Optimization Algorithms

Jesper Christensen, Christophe Bastien, in Nonlinear Optimization of Vehicle Safety Structures, 2016

7.7 Ant colony optimization

Ant colony optimization (ACO) is, as the name suggests, based on the natural behavior of ant colonies and their individual worker ants. When ants forage, they naturally seem to find a “logical” and “effective” route between their nest and the food source; in other words, they seem to determine an optimum route. This observed behavior is the basis of ACO (Bianchi et al., 2008). Imagine two ants walking from the nest to a food source using two different routes. As the ants walk they release pheromones which naturally decay over time. The ant which (randomly) selected the short route will commence the return leg of the journey quicker than the other ant, thus reinforcing the pheromone trace on the shorter route. Other ants will instinctively follow the stronger pheromone path, reinforcing and even adding to this. According to Bianchi et al. (2008), the ACO algorithm contains three major steps that constitute the central optimization loop of the algorithm:

1. Construct Ants Solutions. This is the procedure for which “ants” incrementally and stochastically construct paths, that is, solutions in a wider optimization context;
2. Evaporate Pheromone. This is the process in which pheromone for certain solutions are decreased using “local” information; therefore, this step is also often referred to as local update. This step is pivotal to ensure that the ACO algorithm does not prematurely converge to a single solution;
3. Deamon Actions. This step refers to decisions made based on global information relating to the optimization problem. Note the difference between local in step 2 and global in step 3. Analog to step 2, step 3 is also often referred to as global update.

The three steps highlighted above are repeated until the optimization problem has converged or is otherwise terminated via a prespecified termination condition.

Even though the above description is general and perhaps somewhat abstract in relation to nonlinear topology optimization and the much higher level of details in the previous sections, the basic ideology of the ACO algorithm can be understood. Based on the above, it is clear that significant development and analysis work is required before ACO could be applied to the nonlinear topology optimization problems of concern. It is also clear that it is unlikely for the ACO algorithm to be a “stand-alone solution” to the nonlinear topology optimization problem. It is perhaps more appropriate to see the ACO algorithm as something which could be merged into other structural optimization algorithms, for example as an “evolution” of BESO, that is, an algorithm that could be combined with that presented in Section 7.3. Based on this relatively short review of ACO, it is not feasible to detail a specific list of advantages and disadvantages of ACO for nonlinear topology optimization, quite simply because additional analysis and development work is required. It should, however, be noted that ACO has the potential to be combined with, for example, BESO to overcome the previously stated concerns with respect to the diversity of the search area for BESO algorithms.

> Read full chapter

Introduction to Optimization

Yavuz Eren, ... İlker Üstoğlu, in Optimization in Renewable Energy Systems, 2017

2.4.4 Ant Colony Optimization

Ant colony optimization (ACO), which is developed by Dorigo et al., is also a population-based heuristic approach like PSO technique [80,81]. The motivation of this technique is based on the behavior of the ant colonies for finding food. Each member of the colony makes an effort to find a commonly used path through the food source. In this regard, the ants secrete signaling pheromone to mark their own path for the source. This is the main characteristic of this approach and the follower ants prefer to go on the path with stronger pheromone so that, the change on the density of the pheromone level specifies the selecting probability of each paths.

This technique is developed by observing the food searching efforts of ant clusters. The ant colony follows an organized and smart technique for approaching the food source. The details of the technique is modeled with mathematical tools, and then the approach is transformed into an optimization problem framework to utilize for engineering problems [82,83], such as the search area is defined as graph and the agents (ants) are described as moving point on this graph. As the agents move on the graph, a simulation version of pheromone secreting model is realized with a stochastic approach to mark the most popular paths through the source [84].

Each ant starts to move from randomly selected points on the graph. The connection line from the starting point to the target describe a path and each path is categorized with pheromone level and correlating heuristic value so that, higher of these parameters for a path gives rise to higher the probability an ant prefers this shorter path through the source. The rest of the ants use the pheromone deposited on the path for searching more promising direction through the food target. Then, this iterative procedure goes on until each of the ants finishes their travel for the food and pheromone level is updated on each paths visited by the ants. Consequently, each ant provides a solution and, at least, one path among the solutions should fulfill the termination criterion to finish the all procedure. As the main characteristic of this technique depends on the pheromone level on the route through the food source, the higher the depositing pheromone load, the higher optimality a solution is categorized. Fig. 2.6 depicts the general flowchart of ACO technique which is summarized below.

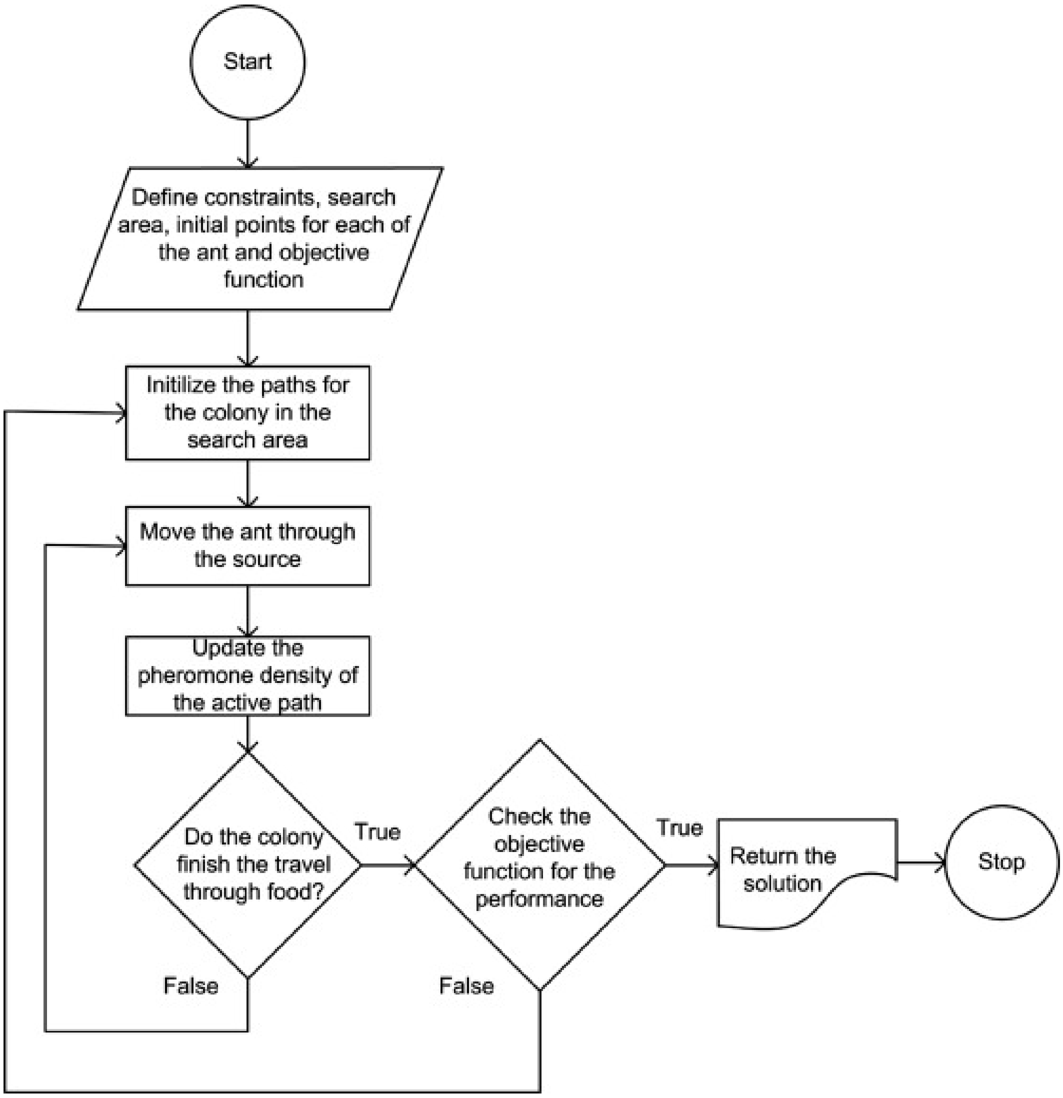
Start
Define constraints, search area, initial points for each of the ant and objective function
Initilize the paths for the colony in the search area
Move the ant through the source
Update the pheromone density of the active path
Do the colony finish the travel through food?
True
False
Check the objective function for the performance
True
False
Return the solution
Stop

Figure 2.6. Flowchart of ACO technique.

Advantages and disadvantages of ACO show similarity with PSO technique, so that the position of the ants are defined on coordinate planes and this technique is also suitable for the problem with parameters lower than three [25]. Moreover, as each ant in the colony makes an effort for following the most feasible path, ACO algorithm is very compatible with parallel implemented solutions which gives the algorithm responsive character.

As increasing global environmental concerns and the needs for more electrical power, alternative energy sources have been penetrated into the grid and ACO technique have been utilized many research activities subjected to these microgrid systems. For controlling of these microgrids and dispatching the energy in the whole system, Colson et al. proposed a supervisory controller based on ACO technique by considering the constraints of fuel availability, economical issues, and environmental problems [85]. They concluded that, the proposed strategy promises an important step for synthesizing heuristic-based autonomous operation strategy for microgrids which provides more sustainable and less operational costs and emissions. Besides that, Pothiya et al. also considered ACO heuristic model for finding more economic dispatch model and concluded that the obtained solution gives a result with higher quality and less computational time compared to the other similar heuristic methods [86].

Another microgrid solution for the above mentioned problem is to utilize wind and solar energy. In this manner, Eroğlu and Seçkiner improved an ACO algorithm for the aim of maximization the expected output power of wind energy source by considering the constraints of turbine locations, wind directions, and some assumption raised in formulation of the problem [87]. The authors tested the algorithm under three different scenarios built by the various cases of the constraints and also compared it with other competitive techniques to reveal the merits of the adopted strategy.

Sizing of hybrid configuration of renewable sources is another current issue. For example, Kefayat et al., utilized ACO and artificial bee colony (ABC) technique for optimal sizing and location of a distributed energy system which is composed of fuel cell, wind, and gasoline sources. The adopted strategy is applied to this hybrid system to satisfy the minimizing the objectives of the power loses, emissions rates, cost of produced energy, and the instability of the voltage. The results obtained expose the effectiveness of the solution techniques compared with the similar evolutionary optimization methods [88]. Real-time application of wind energy based electricity plant considered in [89]. The authors presented a heuristic hybrid optimization solution compromising ACO and PSO techniques for testing the wind-based output power with changeable weather conditions and consequently they obtained a robust forecasting model for the related systems. By considering the solar energy systems under the variable insulation condition, it is extremely important to operate the PV arrays near to MPP in the characteristics of power-voltage. For this aim, Jiang et al. provided a MPP tracking scheme based on ACO technique and they demonstrated the feasibility of this scheme by testing the PV system under different irradiance shading conditions [90], so they concluded that the presenting algorithm ensures for tracking the global MPP in the characteristic as the PV system was operating various shading cases.

> Read full chapter