

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,400

Open access books available

174,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

An Overview of Ant Colony Optimization Algorithms for Dynamic Optimization Problems

Alireza Rezvanian, S. Mehdi Vahidipour and Ali Sadollah

Abstract

Swarm intelligence is a relatively recent approach for solving optimization problems that usually adopts the social behavior of birds and animals. The most popular class of swarm intelligence is ant colony optimization (ACO), which simulates the behavior of ants in seeking and moving food. This chapter aims to briefly overview the important role of ant colony optimization methods in solving optimization problems in time-varying and dynamic environments. To this end, we describe concisely the dynamic optimization problems, challenges, methods, benchmarks, measures, and a brief review of methodologies designed using the ACO and its variants. Finally, a short bibliometric analysis is given for the ACO and its variants for solving dynamic optimization problems.

Keywords: ant colony optimization, dynamic optimization problem, metaheuristics, swarm intelligence, dynamic environments

1. Introduction

Swarm intelligence is a relatively new approach to optimizing problems that typically mimics the social behavior of birds and animals. Ant colony optimization (ACO), which models ant behavior in locating and moving food, is the most prominent type of Swarm intelligence. The early ACO algorithms were applied to discrete optimization problems such as traveling salesman problems, routing, and scheduling. These types of problems consist of a limited number of fixed parameters in stationary environments. However, most real problems have dynamic and time-varying parameters with a continuous range of values. It means the optimal solution is not fixed to changing environments and time. Therefore, the optimization algorithm needs a strategy for tracking optimal solutions. In addition to solving static optimization problems, the ACO algorithm and its variants can also solve dynamic optimization problems.

The rest of this chapter is organized as follows. Section 2 describes the preliminary and background of the work, including the ACO and dynamic optimization problems. Section 3 briefly overviews evolutionary algorithms and ACO for dynamic optimization problems. Section 4 gives a bibliometric analysis of the ACO and its variants for dynamic optimization problems. Finally, Section 5 concludes this chapter.

2. Preliminary and background

In this section, the ACO and dynamic optimization problems are described briefly.

2.1 Ant colony optimization

The Ant Colony Optimization (ACO) algorithm successfully solves various discrete optimization problems. The standard algorithm of ACO was proposed by Dorigo as a multi-agent approach for solving the traveling salesman problem (TSP) [1]. The main idea of the ACO algorithm is inspired by the behavior of colonies of ants seeking food. Usually, at first, ants search their environment randomly for food and return some of the food to their nest once found. They also leave a pheromone on the path they found. The value of pheromones left on their path depends on the quality and size of the food source, which evaporates gradually. The remaining pheromones on a path can persuade other ants to follow the path. After a short time, most ants can follow the shorter path specified by the strongest pheromone. A general pseudocode of the ACO algorithm is given in Algorithm 1.

Algorithm 1. Pseudocode of standard ACO for solving optimization problems.

Ant Colony Optimization Algorithm

Begin

Initialize user parameters of ACO and generate the initial population of the ant colony

While the stopping condition is not met **do**

Position each ant in a starting node

Repeat

For each do

Choose the next node by applying the state transition rule

Apply step-by-step pheromone update

End for

Until every ant has built a solution

Update the best solution

Apply offline pheromone update

End While

End

Through the running of the algorithm, ants first produce different solutions randomly. Then, the solutions are improved by updating the pheromones according to the type of the problem and graph traversal, pheromones set on vertices or edges of the graph. Traversing the edge between two nodes i and j depends on the probability of edge, which is calculated as given follows:

$$p_{ij}^k = \frac{\tau(t)_{ij}^\alpha \eta_{ij}^\beta}{\sum_{j \in N^k(i)} \tau(t)_{ij}^\alpha \eta_{ij}^\beta} \quad (1)$$

where, p_{ij}^k is the probability of traversing from the edge between nodes i and j , $\tau(t)_{ij}^\alpha$ denotes the value of the pheromone trail, η_{ij}^β is the heuristic information, and $N^k(i)$ is the set of nodes that can be visited by ant k . Before updating the pheromones, it is

recommended to perform a local search to improve results. However, the method of updating the pheromones is suggested as given follows [1]:

$$\tau_{ij}(t + 1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

where ρ is the evaporation rate of pheromone and $\Delta\tau_{ij}(t)$ denotes the amount of pheromone trail put on the edge between nodes i and j . That amount is calculated according to the quality of the solutions found by the whole colony, which include the edge between nodes i and j .

2.2 Dynamic optimization problems

A wide range of real-world optimization problems is dynamic, in which the optimal solution(s) to the problems may change over time. An example of such problems includes the Dynamic Vehicle Routing Problem (DVRP) [2], where the online arrival of customers' requests during operation is imposed as the source of dynamism in static vehicle routing, whereby previously found solution(s) may no longer be valid. Another well-known example is the Dynamic Shortest Path Routing Problem (DSPRP) [3], where optimization aims to find the shortest path from a specific source to a specific destination in a changing network environment while minimizing the total cost associated with the path.

As a formal definition [4], a dynamic optimization problem can be defined by a quintuple $\{\Omega, x, \theta, f, t\}$, where Ω is the search space, x denotes a feasible solution in Ω , θ is the system control parameters which determine the solution distribution in the fitness landscape, f represents the static objective function and t denotes the time. To define a dynamic environment, different change types can be defined. Yang and Pelta [5] suggested a framework of eight change types: small step change, large step change, random change, chaotic change, recurrent change, recurrent change with noise, and dimensional change. The purpose of optimization in such problems is no longer just locating the optimal solution(s) but instead tracking the shifting optima over time.

In general, all aspects of science and engineering include optimizing a set of complex problems. In these cases, the optimization objectives, some restrictions, or other elements of the problems may vary over time. In these cases, the optimum solution(s) to the problem are also subject to change. As seen from the examples mentioned earlier, unlike static optimization problems, the goal of optimization in dynamic problems is no longer locating the optimal solution(s) but tracking the trajectories of the optimum(s) over time with a high level of accuracy.

2.2.1 Dynamic optimization problem categories

Weicker et al. [6] proposed a framework for dynamic optimization problems based on coordinate transformations (i.e., change in the position of the optima) and fitness rescaling (i.e., change in the value of the objective function of the optima). Eberhart et al. [7] classified dynamic environments into three major types: Type I, Type II, and Type III. In type I environments, the position of the optima changes over time; however, its objective function's value remains unchanged. In type II, the position of the optima does not change but its value varies over time. Finally, type III, where both the position and the value of the optima change over time.

Angeline [8] classified different dynamic environments based on the trajectories of peaks into three categories: linear, circular, and random. According to Duhain et al. [9], dynamic environments can be categorized based on how severe the changes are over time and space. They call them behavioral classes, including progressively changing, abruptly changing, and chaotic environments. Progressively changing environments have frequent temporal changes but small spatial changes, and they change gradually and smoothly. In abruptly changing environments, the changes happen rarely, but the spatial alterations are severe. Finally, if both the spatial and temporal severity of the dynamic environment is high, the behavior of the environment is chaotic. Moreover, they combined their behavioral classification with Eberhart et al. [7] and Angeline's [8] types and defined 27 kinds of dynamic environments based on the direction, the trajectory, and the severity of the changes that the environments undergo. A more comprehensive study of dynamic optimization problems with a wider scope can be found in the literature [10–15].

2.2.2 Optimization challenges in dynamic optimization problems

The purpose of optimization in dynamic optimization problems is no longer to locate the stationary optimal solution(s) but to track the trajectories of the optimum (s) over time as accurately as possible [16–18]. This situation possesses additional challenges that should be addressed properly to obtain promising results. Reacting to changes plays a critical role in maintaining the performance of optimization algorithms in dynamic environments. The first step toward reacting to changes is to detect them in the first place and then adopt a suitable strategy to deal with them.

Richter [19] discussed two major types of change detection, referred to as population-based detection and sensor-based detection. In population-based detection, statistical hypothesis testing is performed at each generation to see whether the alteration in the fitness of the individuals was not due to their convergence. In sensor-based detection, some measurements, so-called “fitness landscape sensors,” are placed either randomly or on a regular pattern into the landscape. At every generation, the environment changes if any of the sensors detect an altered fitness value. As mentioned earlier, the two mechanisms have their own merits and limits. Population-based detection does not impose any additional fitness function evaluations on the algorithms, while sensor-based detection, on the other hand, can eliminate elaborate statistical hypothesis testing procedures. Many existing works in the literature reevaluate the previous best solutions to detect environmental changes [20] ((References)). This strategy can be regarded as a combination of both mentioned change detection types, as reevaluating previous best solutions can be considered a kind of sensor-based detection where the previous best individuals dynamically allocate the distribution of the sensors in the landscape.

For optimization algorithms to hopefully advance the search process in dynamic environments, two issues must be addressed properly. The first issue arising from the change in the environment is outdated information, causing the quality of previous solutions to be compromised. This has a deleterious effect on the performance of the optimization and can mislead the search process toward the wrong positions in the landscape. This problem can be eliminated by reevaluating past information or replacing it with current information. The second issue is diversity loss, which is much more serious than the other challenges. This issue arises when the population of a population-based heuristic converges to a peak. When the environment changes, if the peak goes out of the population's spatial range, the function evaluations required

for a partially converged population to re-diversify and re-converge to the shifted peak are quite detrimental to the performance [21]. Therefore, maintaining diversity is essential for population-based methods to adapt the imposed changes.

2.2.3 Dynamic optimization problems benchmarks

Recently, there has been a growing interest in the resolution of synthetic dynamic optimization problems due to the ability of these models to approximate a variety of real-world situations closely. The first and most widely used real-valued synthetic dynamic optimization test suite in the literature is the Moving Peaks Benchmark (MPB) proposed by Branke [22], which is highly regarded due to its configurability. The baseline function for MPB is specified as given in the following equation:

$$f_t(\mathbf{x}) = \max_{i \in \{1, \dots, m\}} \frac{H_t(i)}{1 + W_t(i) \sum_{j=1}^D (x_t(j) - X_t(i,j))^2}, \quad (3)$$

where m is the number of peaks, $x_t(j)$ is the j^{th} parameter of solution x in a d -dimensional search space, $X_t(i,j)$ is the j^{th} parameter of the center of i^{th} peak in the t^{th} time, $H_t(i)$ is the height of i^{th} peak, and $W_t(i)$ is the width of the i^{th} peak the time t .

The Gaussian Peak Bench (GPB) [23] provides another method for generating moving peaks. In the Generalized moving peaks benchmark (GMPB) [24], the baseline function generates landscapes created by collecting some components. The generated modules by GMPBs can differ from symmetric to asymmetric, unimodal to multimodal, flat to highly irregular, and circular contour lines to highly ill-conditioned.

2.2.4 Evaluation measures

Several measurement metrics are proposed to evaluate the optimization algorithms' performance and different aspects of the methods for dynamic optimization problems. For instance, Weicker [25] introduced accuracy, stability, and reactivity as key optimization characteristics in dynamic environments. Alba et al. [26] proposed $\beta_{\text{degradation}}$ to measure the degradation of optimization algorithms in dynamic environments.

A more general approach to measuring an attribute of dynamic optimization problems is proposed in Ref. [27], which calculates an attribute of a population as the area below the curve. Ayvaz et al. [28] proposed a dissimilarity factor, which evaluates the ability of an optimization algorithm to recover after a change. Recently, Kordestani et al. [29] proposed two measures to compare algorithms in dynamic environments. The first measure is alpha-accuracy, which calculates the degree of closeness of the optimization error to a certain expected level during the optimization process. The second measure is fitness adaptation speed which evaluates the average speed of algorithms in adapting to environmental changes. Despite the number of performance measures in the literature, most authors have used offline error for measuring their proposed algorithms. This is because they can compare their methods with other peer algorithms. This measure is defined as given follows [17, 29–31]:

$$E_O = \frac{1}{T} \sum_{t=1}^T e_t^* \quad (4)$$

Name	References	Measure
Accuracy	[25, 27]	The goodness of the best-found solution in the current population
Stability	[25, 27]	The impact of changes in the environment on the optimization accuracy
Reactivity	[25, 27]	The ability of the algorithm to react quickly to changes
Offline Error	[35]	The average fitness error of the best-found solution at every point in time
Best Error Before Change	[32, 34, 36]	The average fitness error of the best-found solution over the environmental changes
Fitness Degradation	[26]	The degradation in the quality of solutions as changes in the landscape occur
Area Below Curve	[27, 37]	An attribute of the population is the area below the curve
Dissimilarity Factor	[28]	The ability of an optimization algorithm to recover after a change
Alpha-Accuracy	[20, 29]	The degree of closeness of the optimization error to a certain expected level during the optimization process
Fitness Adaptation Speed	[20, 29]	The average speed of the algorithms in adapting to the environmental changes

Table 1. Summary of the performance measures for dynamic optimization problems.

where T is the maximum number of function evaluations and e_t^* is the minimum error gained by the optimization algorithm at the t^{th} fitness evaluation. Another measure used in Ref. [32] has also been termed offline error as the best error before the change. This measure is slightly different from the previously mentioned measure in Eq. (4) and may become a source of confusion when comparing different methods. This measure, first proposed in Ref. [33] as an accuracy criterion and then named the best error before change by Nguyen et al. [12], is calculated as the average of the minimum fitness error achieved by the algorithm at the end of each period right before the moment of change, as given follows [34]:

$$E_B = \frac{1}{K} \sum_{k=1}^K (h_k - f_k) \quad (5)$$

where f_k is the best solution obtained by an algorithm just before the k^{th} change happens, h_k is the optimum value of the k^{th} environment, and K is the total number of environments—the only difference between Eqs. (4) and (5) are the time taken to record the error gained by the algorithm, where in Eq. (4), the error is recorded every time the function is evaluated, and in Eq. (5), it is recorded right before the change. **Table 1** summarizes some of the existing performance measures in the literature.

3. Related works

3.1 Evolutionary computation for dynamic optimization problems

Since exact algorithms are impractical in dynamic environments, stochastic optimization techniques have become popular. Among them, evolutionary computation

(EC) techniques have attracted much attention due to their potential for solving complex optimization problems. Nevertheless, evolutionary computation methods should undergo certain adjustments to work well when applied to dynamic optimization problems. Diversity loss is the most severe challenge to evolutionary computation methods in dynamic optimization problems. This issue arises from the tendency of individuals to converge on a single optimum. As a result, when the global optimum is shifted away, the number of function evaluations (FEs) required for a partially converged population to relocate the optimum is quite harmful to performance.

Over the years, researchers have proposed various techniques to improve traditional evolutionary computation methods for solving dynamic optimization problems. According to Reference *Nguyena et al. [12]*, the existing proposals can be categorized into the following six approaches:

1. Increasing the diversity after detecting a change in the environment
2. Maintaining diversity during the optimization process
3. Employing memory schemes to retrieve information about previously found solutions
4. Predicting the location of the next optimal solution(s) after a change is detected
5. Making use of the self-adaptive mechanisms of evolutionary computations
6. Using multiple sub-populations to handle separate areas of the search space concurrently

As mentioned, the multi-population (MP) approach effectively handles dynamic optimization problems, especially in multimodal fitness landscapes. The success of MP methods can be attributed to the following three reasons [12]:

1. Population diversity can still be maintained globally even when populations search within different subareas of the fitness landscape.
2. It is possible to locate and track multiple changing optima simultaneously. This feature can facilitate tracking of the global optimum, given that one of the being tracked local optima may become the new global optimum when environmental changes occur.
3. Extending any single-population approach, e.g., diversity increasing/maintenance schemes, memory schemes, and adaptive schemes, is easy.

There are several studies in the literature on dynamic optimization problems in general. For instance, *Jin and Branke [13]* addressed and discussed various types of uncertainty in evolutionary optimization. *Cruz et al. [11]* have contributed to the advancement of knowledge and relevance of the field by accomplishing a twofold task: (1) They have established a public web repository of a large number of relevant references about the topic over the past decade and then categorized the references based on the type of publications, type of dynamism, methods for solving dynamic optimization problems, performance measures, applications and publication year;

(2) Afterward, they have performed an overview of the research carried out on dynamic optimization problems based on the collected repository. Nguyen et al. [12] have provided an in-depth survey of evolutionary optimization in dynamic environments. The study examined state-of-the-art academic research from four different points of view: benchmark problems, performance measures, methodology, and theory.

Yazdani et al. [14, 15] in two parts of survey papers, represented a review of research studies regarding single-objective unconstrained continuous dynamic optimization problems. Since an efficient dynamic optimization algorithm consists of several parts to cope with dynamic optimization problems, they tried to provide a comprehensive taxonomy to identify the parts of dynamic optimization algorithms. In the second part of this survey, they gave an overview of dynamic optimization problem benchmarks and performance measures.

Another study in the literature is provided by Moser et al. [38]. In their work, the authors have given an overview of the available approaches in the literature for solving MPB. They have examined various methods for the MPB in four groups: Evolutionary Algorithms, Swarm Intelligence Algorithms, Hybrid Approaches, and Other Approaches. Although their work is very valuable in studying and summarizing different methods for solving the MPB, it does not provide categorical information on how different methods work and which mechanisms can improve the performance of different methods. In addition, it does not explain the reasons for the superiority of specific approaches. Therefore, we believe it is necessary to provide a well-structured survey of available methods for solving the MPB to extract their strengths and weaknesses and open the field to develop new efficient approaches to address dynamic optimization problems.

3.2 ACO for dynamic optimization problem

Ant colony optimization is a candidate suited for solving dynamic optimization problems due to its ability to adapt to changing environments. It searches for optimal solutions in time-varying environments. Basically, this is accomplished through the use of pheromones. These are chemical signals deposited by ants as they move through the environment. Also, because of the evaporating nature of pheromones, ants can quickly find promising regions of the solution space. They can converge on proper solutions even as the problem changes over time.

In the following, brief overviews of the studies regarding ACO methods for the dynamic optimization problem are presented. [39] is the first study of ant colonies for optimization in a continuous and dynamic environment. The Dynamic Hybrid Continuous Interacting Ant Colony (DHCIAC), as a hierarchical algorithm focused on the communication behaviors of the ants and hybridized an interacting ACO for collecting information about the environment with a Nelder-Mead local search [40] for improving the solutions. The interacting algorithm was originally designed for a static environment that used several communication channels with different characteristics. The first channel is stigmergic, an indirect coordination mechanism between agents and their environments. Ants move around pheromone regions due to their role. The second channel is the direct channel where ants can communicate through messages. Each ant has a stack of messages and can send messages to the other ants. Also, based on a parameter, the ants can follow a new solution or boost the current one. According to the hierarchical framework of the

algorithm and its channels, the DHCIAC can propagate information regarding the changing environment. Finally, an efficient local search can improve results. The authors also developed two versions of DHCIAC with different mechanisms for finding dynamic optimal. DHCIAC_track uses an enhanced local search to track the optimal evolution, while DHCIAC_find uses a classical Nelder-Mead local search to find the optimal solution whenever the environment changes.

The charged ants for continuous dynamic optimization are represented by Tfaili et al. [41]. The authors proposed assigning a repulsive electric charge to each ant. This charge prevents ants from getting close to maintaining a certain level of diversity. In this algorithm, the ants initially distribute in the environment randomly. Then, the new continuous solutions are generated based on a weighted Gaussian distribution as pheromones. The charged values of ants can be positive or negative based on the quality of the best solutions. The algorithm works with a limited and fixed population, and each ant can produce a solution similar to ACO [42]. As each interaction proceeds, the better new solutions replace the older and worse solutions.

As an application of the ant colony optimization, Montemanni et al. [43] presented the dynamic version of the ACO for the Vehicle Routing Problem (VRP), where the problem parameters change over time. Dynamic VRP aims to find an optimal route plan for a fleet of vehicles. This route plan considers changes in parameters such as customer demands, vehicle availability, and traffic conditions. Three modifications are incorporated into the ACO algorithm to cope with this problem, including local search, ant behavior, and pheromone updating. With local search, the quality of the solutions can be improved as the parameters of the problem change over time. Ant behavior modification can improve ants' behavior to adjust the changing parameters. Finally, by pheromone updating modification, pheromone updating adjusted the parameters over time.

Korosec et al. suggested a differential ant-Stigmergy algorithm for dynamic optimization problems [44]. Stigmergy is commonly used in multi-agent systems as a kind of indirect cooperation or coordination between agents and the environment. An optimization process boosts the best current solution by creating a proper path. The path is based on graph search using pheromones assigned to a Cauchy distribution. Then a path from a source node to a destination node is formed based on choosing each ant from another ant according to its probability. This process is repeated until reaching a destination node. By finding paths, the paths are evaluated, and the pheromones are updated.

In [45], the authors introduced a dynamic strategy to adapt the parameters of ACO through the search process dynamically. The strategy helps the algorithm adapt to environmental changes and prevent falling into local optima. Besides, they utilize a mechanism based on entropy to balance exploration and exploitation in the search process to improve the quality of solutions. Finally, the algorithm is applied to solving the TSP and VRP.

In [46], Zhou et al. represented an improved version of ACO, which incorporates two significant mechanisms parameter adaptation and dynamic hybridization. The parameter adaptation mechanism encompasses adapting the parameters of the ACO algorithm dynamically based on the features of the problem being solved. This mechanism better guides the algorithm to balance the search process's exploration and exploitation. The dynamic hybridization mechanism hybridizes the ACO algorithm with another metaheuristic algorithm, such as Genetic Algorithm or Simulated Annealing, depending on the growth of the

Reference(s)	Problem
[49]	Dynamic Electric Vehicle Dispatch Optimization
[50]	Dynamic Chemical Processes
[51]	Dynamic Cold Chain Logistics Scheduling
[52]	Dynamic Distribution Network
[53]	Dynamic Knapsack Problem
[54, 55]	Dynamic Load Balancing
[56]	Dynamic Location Routing Problem
[48]	Dynamic Quadratic Assignment Problem
[57]	Dynamic Railway Junction Rescheduling Problem
[58]	Dynamic Traveling Salesman Problem
[59, 60]	Dynamic Vehicle Routing Problem

Table 2.
Summary of ACO applications for dynamic optimization problems.

search process. This idea allowed the algorithm to avoid local optima and keep the diversity of solutions.

Brest et al. [47] introduced two population-based algorithms based on differential evolution and ACO with continuous variables: the self-adaptive differential evolution algorithm (jDE) and the differential ant-stigmergy algorithm (DASA). These algorithms use the behavior of ants to guide their search process, and ants can communicate using pheromones in the environment to identify the optimal solution. They focused on comparing the performances of these two algorithms.

In [48], Guntzsch et al. proposed a population-based ant colony optimization (PBACO) as an improved version of the ACO algorithm that employs several colonies to search for solutions. In the PBACO, a mechanism is designed for handling changes in the problem environment over time. Also, an algorithm is devised with multi-population in which each population is responsible for searching for solutions in a different time of the environment. Then, the best solution for each population is shared among the populations to allow for information exchange and exploration of different regions of the search space. To prevent falling into local solutions, a migration strategy can also be used to maintain diversity among the populations. Furthermore, some applications of ACO for dynamic optimization problems have been tabulated in **Table 2**.

4. Bibliometric analysis of ACO for dynamic optimization problems

The publication on Scopus by the query for title, abstract, and keywords with this query: (“ACO” OR “ant colony”) AND (“dynamic optimization” OR “dynamic optimization” OR “dynamic environment” OR “uncertain environment” OR “time-varying environment” OR “dynamic problem”)) have been reviewed. The resulting list of publication years from 2003 to 2023 is plotted in **Figure 1**. **Figure 1** shows that the number of publications on the topic of dynamic optimization using the ACO is gradually increasing.

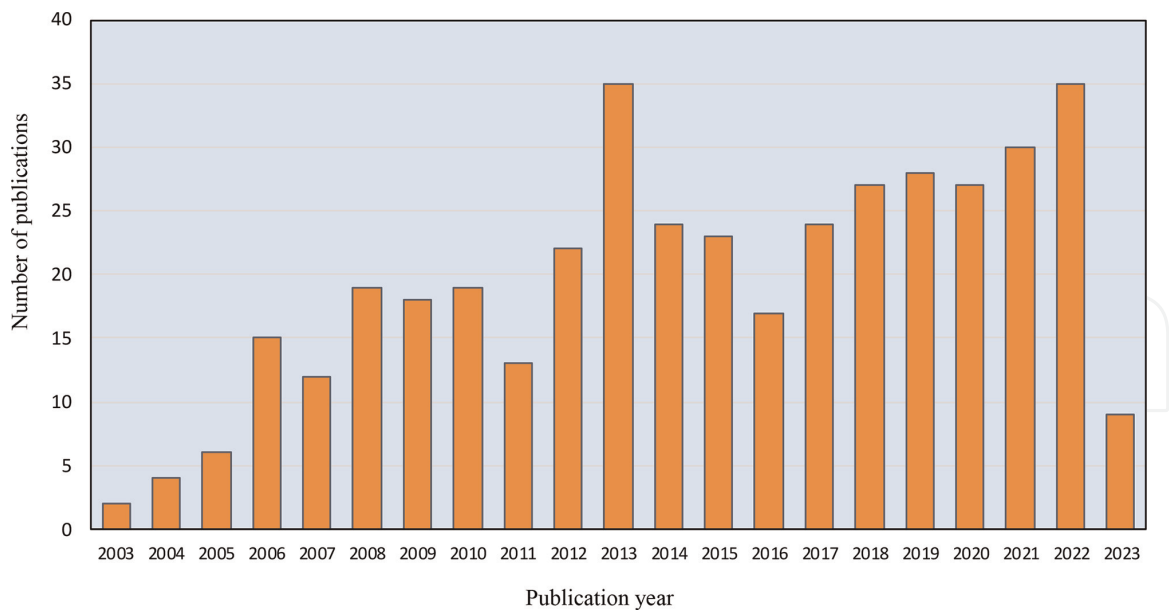


Figure 1.
 Distribution of the number of publications that used ACO for dynamic optimization problems.

Furthermore, the top 10 countries with the most publications are plotted in **Figure 2**. As shown in **Figure 2**, most publications are published by researchers from China (34%).

The top 10 universities with the most publications are plotted in **Figure 3**. Looking at **Figure 3**, most publications have authors from De Montfort University, UK, having 16 records.

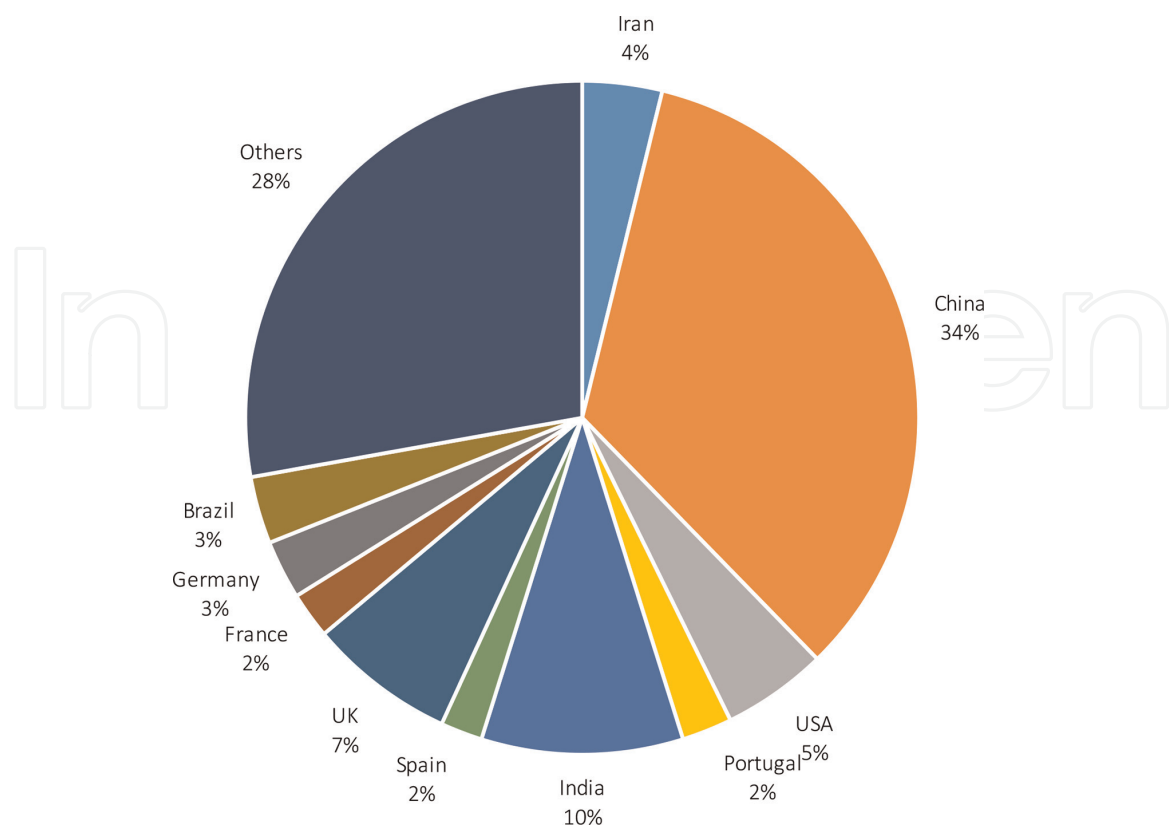


Figure 2.
 Distribution of the top 10 countries with the most publications.

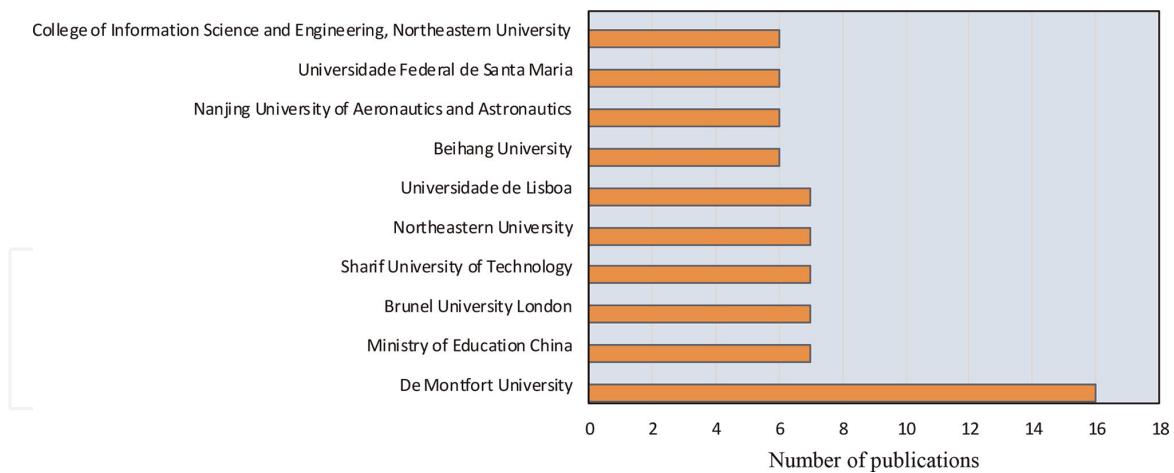


Figure 3. Distribution of the top 10 universities with the most publications applying ACO for solving dynamic optimization problems.

5. Conclusions

As one of the most popular metaheuristics, the ant colony optimization (ACO) algorithm solves various static and dynamic optimization problems. Since most real optimization problems have dynamic and time-varying parameters with a continuous range of values, several ant colony optimization algorithms were developed to deal with dynamic optimization problems. This chapter presents a brief overview of ACO and its variants for dynamic optimization problems. Furthermore, the definition, challenges, benchmarks, measurement metrics, and a short bibliometric analysis have been provided in this chapter. In the following years, the application of ACO and its variant is expected to increase gradually for solving dynamic optimization problems.

Research can be facilitated in the future by the use of adaptive ACO algorithms, in which they can adjust the search strategy based on the current state of the environment, in order to solve dynamic optimization problems, as well as different strategies for updating pheromones, such as decaying pheromones, reinforcing pheromones, and balancing exploration with exploitation by adjusting search parameters dynamically using learning automata.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Conflict of interest

All authors declare that they have no conflict of interest.

Authorship change

The author names will be published exactly as they appear on the initial submission.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

IntechOpen

Author details

Alireza Rezvanian^{1*}, S. Mehdi Vahidipour² and Ali Sadollah³


1 Department of Computer Engineering, University of Science and Culture, Tehran, Iran

2 Computer Engineering Department, Faculty of Electrical and Computer Engineering, University of Kashan, Kashan, Iran

3 Department of Mechanical Engineering, University of Science and Culture, Tehran, Iran

*Address all correspondence to: rezvanian@usc.ac.ir

IntechOpen

© 2023 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Dorigo M, Di Caro G. Ant colony optimization: A new meta-heuristic. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). Washington, DC, USA: IEEE; 1999. pp. 1470-1477
- [2] Pillac V, Gendreau M, Guéret C, Medaglia AL. A review of dynamic vehicle routing problems. *European Journal of Operational Research*. 2013; **225**(1):1-11
- [3] Yang S, Cheng H, Wang F. Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in Mobile ad hoc networks. *IEEE Transactions on Systems, Man, and Cybernetics. C Applications Review*. 2010; **40**(1): 52-63
- [4] Novoa-Hernández P, Corona CC, Pelta DA. Efficient multi-swarm PSO algorithms for dynamic environments. *Memetic Computing*. 2011; **3**:163-174
- [5] Li C, Yang S, Pelta DA. Benchmark Generator for the IEEE WCCI-2012 Competition on Evolutionary Computation for Dynamic Optimization Problems. UK: Brunel University; 2011
- [6] Weicker K. An analysis of dynamic severity and population size. In: Lecture Notes in Computer Science, 1917, Parallel Problem Solving from Nature-PPSN VI: 6th International Conference, Paris, France, September 18-20, 2000, Proceedings. 2000. pp. 159-168
- [7] Eberhart RC and Shi Y. Tracking and optimizing dynamic systems with particle swarms. In: *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, Seoul, Korea (South). 2001. pp. 94-100.
- [8] Angeline P. Tracking extrema in dynamic environments. In: *Evolutionary Programming VI. Lecture Notes in Computer Science, Vol 1213*. Berlin, Heidelberg: Springer; 1997. pp. 335-345
- [9] Duhain JG, Engelbrecht AP. Towards a more complete classification system for dynamically changing environments. In: *2012 IEEE Congress on Evolutionary Computation*. Brisbane, QLD, Australia: IEEE; 2012. pp. 1-8
- [10] Branke J. *Evolutionary Optimization in Dynamic Environments*. New York, NY: Springer; 2002
- [11] Cruz C, González JR, Pelta DA. Optimization in dynamic environments: A survey on problems, methods and measures. *Soft Computing*. 2010; **15**(7): 1427-1448
- [12] Nguyena TT, Yangb S, Brankec J. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*. 2012; **6**: 1-24
- [13] Jin Y, Branke J. Evolutionary optimization in uncertain environments—a survey. *IEEE Transactions on Evolutionary Computation*. 2005; **9**(3): 303-317. DOI: 10.1109/TEVC.2005.846356
- [14] Yazdani D, Cheng R, Yazdani D, Branke J, Jin Y, Yao X. A survey of evolutionary continuous dynamic optimization over two decades—Part a. *IEEE Transactions on Evolutionary Computation*. 2021; **25**(4):609-629
- [15] Yazdani D, Cheng R, Yazdani D, Branke J, Jin Y, Yao X. A survey of evolutionary continuous dynamic optimization over two decades—Part B.

IEEE Transactions on Evolutionary Computation. 2021;25(4):630-650

[16] Kordestani JK, Rezvanian A, Meybodi MR. An efficient oscillating inertia weight of particle swarm optimisation for tracking optima in dynamic environments. *Journal of Experimental & Theoretical Artificial Intelligence*. 2015;28:1-13. DOI: 10.1080/0952813X.2015.1020521

[17] Kordestani JK, Mirsaleh MR, Rezvanian A, Meybodi MR. *Advances in Learning Automata and Intelligent Optimization*. Cham, Switzerland: Springer; 2021

[18] Kordestani JK, Rezvanian A, Meybodi MR. CDEPSO: A bi-population hybrid approach for dynamic optimization problems. *Applied Intelligence*. 2014;40(4):682-694. DOI: 10.1007/s10489-013-0483-z

[19] Richter H. Detecting change in dynamic fitness landscapes. In: *IEEE Congress on Evolutionary Computation*. 2009. pp. 1613–1620. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4983135 [Accessed: October 30, 2012]

[20] Kazemi Kordestani J, Razapoor Mirsaleh M, Rezvanian A, Meybodi MR. An overview of multi-population methods for dynamic environments. *Advances in Learning Automative Intelligence Optimization*. 2021;208: 253-286

[21] Blackwell T. Particle swarm optimization in dynamic environments. *Evolutionary Computation in Dynamic and Uncertain Environments*. 2007;51: 29-49

[22] Branke J. Memory enhanced evolutionary algorithms for changing optimization problems. In: *Proceedings*

of the 1999 Congress on Evolutionary Computation. Washington, DC, USA: IEEE; 1999. pp. 1875-1882

[23] Grefenstette JJ. Evolvability in dynamic fitness landscapes: A genetic algorithm approach. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Washington, DC, USA. 1999. pp. 2031-2038

[24] Yazdani D, Omidvar MN, Cheng R, Branke J, Nguyen TT, Yao X. Benchmarking continuous dynamic optimization: Survey and generalized test suite. *IEEE Transactions on Cybernetics*. 2020;52(5):3380-3393

[25] Weicker K. Performance measures for dynamic environments. In: *Parallel Problem Solving from Nature—PPSN VII*. Berlin, Heidelberg: Springer; 2002. pp. 64-73. Accessed: May 22, 2014 [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-45712-7_7

[26] Alba E, Sarasola B, Di Chio C. Measuring fitness degradation in dynamic optimization problems. In: *Applications of Evolutionary Computation*, in *Lecture Notes in Computer Sciences*. Vol. 6024. Berlin / Heidelberg: Springer; 2010. pp. 572-581. Accessed: Oct. 13, 2012 [Online]. Available: <http://www.springerlink.com/index/716778080534516x.pdf>

[27] Sarasola B, Alba E, Alba E. Quantitative performance measures for dynamic optimization problems. In: *Metaheuristics for Dynamic Optimization*, in *Studies in Computational Intelligence*. Vol. 433. Berlin/Heidelberg: Springer; 2013. pp. 17-33. Accessed: Oct. 13, 2012 [Online]. Available: <http://www.springerlink.com/index/A151765314542484.pdf>

- [28] Ayvaz D, Topcuoglu HR, Gurgun F. Performance evaluation of evolutionary heuristics in dynamic environments. *Applied Intelligence*. 2012;**37**(1): 130-144. DOI: 10.1007/s10489-011-0317-9
- [29] Kordestani JK, Rezvanian A, Meybodi MR. New measures for comparing optimization algorithms on dynamic optimization problems. *Natural Computing*. 2019;**18**:705-720
- [30] Blackwell T, Branke J. Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation*. 2006;**10**(4):459-472. DOI: 10.1109/TEVC.2005.857074
- [31] Ranginkaman AE, Kordestani JK, Rezvanian A, Meybodi MR. A note on the paper 'A multi-population harmony search algorithm with external archive for dynamic optimization problems' by Turkey and Abdullah. *Information Sciences*. 2014;**288**:12-14
- [32] Li C, Yang S. A general framework of multipopulation methods with clustering in undetectable dynamic environments. *IEEE Transactions on Evolutionary Computation*. 2012;**16**(4):556-577. DOI: 10.1109/TEVC.2011.2169966
- [33] Trojanowski K and Michalewicz Z. Searching for optima in non-stationary environments. In: *Proceedings of the 1999 Congress on Evolutionary Computation, 1999, (CEC 99)*. 1999. pp. 1-5. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=785498 [Accessed: October 13, 2012]
- [34] Yang S, Li C. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation*. 2010;**14**(6): 959-974. DOI: 10.1109/TEVC.2010.2046667
- [35] Branke J, Schmeck H. Designing evolutionary algorithms for dynamic optimization problems. *Advances in Evolution Computer Theory Applications*. 2003:239-262
- [36] Li C and Yang S. Fast multi-swarm optimization for dynamic optimization problems. In: *Fourth International Conference on Natural Computation, 2008, (ICNC'08)*. 2008. pp. 624-628. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4668051 [Accessed: October 13, 2012]
- [37] Alba E and Sarasola B. ABC, a new performance tool for algorithms solving dynamic optimization problems. In: *2010 IEEE Congress on Evolutionary Computation (CEC)*. 2010. pp. 1-7. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5586406 [Accessed: October 13, 2012]
- [38] Moser I, Chiong R. Dynamic function optimization: The moving peaks benchmark. *Metaheuristics Dynamic Optimization*. 2013;**433**:35-59
- [39] Dréo J, Siarry P. An ant colony algorithm aimed at dynamic continuous optimization. *Applied Mathematics and Computation*. 2006;**181**(1):457-467. DOI: 10.1016/j.amc.2005.12.051
- [40] Nelder JA, Mead R. A simplex method for function minimization. *The Computer Journal*. 1965;**7**(4): 308-313
- [41] Tfaili W, Siarry P. A new charged ant colony algorithm for continuous dynamic optimization. *Applied Mathematics and Computation*. 2008;**197**(2):604-613. DOI: 10.1016/j.amc.2007.08.087

- [42] Socha K, Dorigo M. Ant colony optimization for continuous domains. *European Journal of Operational Research*. 2008;**185**(3):1155-1173
- [43] Montemanni R, Gambardella LM, Rizzoli AE, Donati AV. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*. 2005;**10**:327-343
- [44] Korosec P, Silc J. The differential ant-stigmergy algorithm applied to dynamic optimization problems. In: *2009 IEEE Congress on Evolutionary Computation*. Trondheim, Norway: IEEE; 2009. pp. 407-414
- [45] Chen J, You X-M, Liu S, Li J. Entropy-based dynamic heterogeneous ant colony optimization. *IEEE Access*. 2019;**7**:56317-56328
- [46] Zhou X, Ma H, Gu J, Chen H, Deng W. Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. *Engineering Applications of Artificial Intelligence*. 2022;**114**:105139. DOI: 10.1016/j.engappai.2022.105139
- [47] Brest J, Korošec P, Šilc J, Zamuda A, Bošković B, Maučec MS. Differential evolution and differential ant-stigmergy on dynamic optimisation problems. *International Journal of Systems Science*. 2013;**44**(4):663-679
- [48] Guntsch M, Middendorf M. Applying population based ACO to dynamic optimization problems. In: *Ant Algorithms: Third International Workshop, ANTS 2002 Brussels, Belgium, September 12–14, 2002 Proceedings 3*. Berlin, Heidelberg: Springer; 2002. pp. 111-122
- [49] Shi L, Zhan Z-H, Liang D, Zhang J. Memory-based ant colony system approach for multi-source data associated dynamic electric vehicle dispatch optimization. *IEEE Transactions on Intelligent Transportation Systems*. 2022; **23**(10):17491-17505
- [50] Xu B, Cheng W, Qian F, Huang X. Self-adaptive differential evolution with multiple strategies for dynamic optimization of chemical processes. *Neural Computing and Applications*. 2019;**31**:2041-2061
- [51] Wu L-J, Shi L, Zhan Z-H, Lai K-K, Zhang J. A buffer-based ant colony system approach for dynamic cold chain logistics scheduling. *The IEEE Transactions on Emerging Topics in Computational Intelligence*. 2022; **6**(6):1438-1452
- [52] Liu J, Huang T, Duan X. Application of improved ant Colony algorithm in optimal planning of dynamic distribution network. In: *Journal of Physics: Conference Series*. IOP Publishing; 2020. p. 032068
- [53] Randall M. A dynamic optimisation approach for ant colony optimisation using the multiple knapsack problem. In: *The Second Australian Conference on Artificial Life*. Sydney: World Scientific; 2005
- [54] Sim KM, Sun WH. Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*. 2003;**33**(5):560-572
- [55] Muteeh A, Sardaraz M, Tahir M. MrLBA: Multi-resource load balancing algorithm for cloud computing using ant colony optimization. *Cluster Computing*. 2021;**24**(4):3135-3145
- [56] Gao S, Wang Y, Cheng J, Inazumi Y, Tang Z. Ant colony optimization with

clustering for solving the dynamic location routing problem. *Applied Mathematics and Computation*. 2016; **285**:149-173

[57] Eaton J, Yang S, Mavrovouniotis M. Ant colony optimization with immigrants schemes for the dynamic railway junction rescheduling problem with multiple delays. *Soft Computing*. 2016;**20**:2951-2966

[58] Silva CA, Runkler TA. Ant colony optimization for dynamic traveling salesman problems. *ARCS 2004–Organic Pervasive Computing*. 2004:259-266

[59] Xiang X, Qiu J, Xiao J, Zhang X. Demand coverage diversity based ant colony optimization for dynamic vehicle routing problems. *Engineering Applications of Artificial Intelligence*. 2020;**91**:103582

[60] Xiang X, Tian Y, Zhang X, Xiao J, Jin Y. A pairwise proximity learning-based ant colony algorithm for dynamic vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*. 2021;**23**(6): 5275-5286

IntechOpen