



A Review on Ant Colony Optimization for Scheduling Problems

Meegan Scariah*

Department of Computer Science, Mar Athanasius College of Engineering, Kerala, India

*Corresponding Author's E-mail: meeganscariah1997@gmail.com

Received: 02-November-2023, Manuscript No. JRJESTI-24-119116; **Editor assigned:** 06-November-2023, PreQC No. JRJESTI-24-119116 (PQ); **Reviewed:** 20-November-2023, QC No. JRJESTI-24-119116; **Revised:** 02-January-2024, Manuscript No. JRJESTI-24-119116 (R); **Published:** 08-January-2024, DOI: 10.14303/ 2315-5663.2024.113

Abstract

Ant colony optimization is a swarm intelligence approach that has proved to be useful in solving several classes of discrete and continuous optimization problems. One set, called scheduling problems, is extremely important both to academics and to practitioners. This paper describes how the current literature uses the ACO approach to solve scheduling problems. An analysis of the literature allows one to conclude that ACO is a hugely viable approach to solve scheduling problems. On the basis of the literature review, we were not only able to derive certain guidelines for the implementation of ACO algorithms but also to determine possible directions for future research.

Keywords: Scheduling, MILP, Single machine, Periodic maintenance, Ant colony

INTRODUCTION

Ant Colony Optimization (ACO) is a meta-heuristic approach proposed by Colnani et al., and improved in later research (e.g., see Dorigo et al., for the ant colony system and Stutzle and Hoos, for the max-min ant system). The behavior common to all approaches involving ant-based algorithms lies in the mimicry of the behavior used by “real” ants to find the optimal path between their nest and a food source (Zhuo et al., 2007).

The earlier application of ACO was to solve the well-known NP-hard traveling salesman problem (Colnani et al., Dorigo et al., Dorigo and Gambardella). In this problem, there is a graph in which each node corresponds to a city, and the arcs correspond to the distances between cities (Zhou et al., 2009). The problem consists of obtaining a minimal tour (sequence of cities) length that contains all the nodes (Zhou et al., 2007).

Several studies have applied ACO to solve different discrete and continuous optimization problems, such as vehicle

routing, quadratic assignment problems and graph coloring (Zhou et al., 2010). Dorigo and Stutzle reported more than 30 problems where ACO-based algorithms have been used successfully (Yoshikawa et al., 2006).

One of these applications involves scheduling problems (Ying et al., 2003). With the significance of these problems recognized because of their impact on real environments and their academic relevance (e.g., Pinedo), “scheduling problems” are a huge set of problems, and are mostly NP-Hard, that try to deal with a simple question: Given a set of jobs, a set of resources, a set of constraints, and an objective function, how should the jobs be allocated to the resources?

Answering this question, however, usually requires complex and/or time-costly procedures (Ying et al., 2007). The great advantage in using a meta-heuristic such as ACO to obtain near-optimal solutions is that the time required to solve the problem is usually acceptable even though the 100 (Yagmahan et al., 2008).

This paper aims at reviewing and classifying published studies that use ACO to solve scheduling problems, and it focuses on the four classical manufacturing environments (single machine, parallel machine, flowshop and jobshop) (Xu et al., 2012). Different scheduling problems, such as service scheduling (e.g., see Gutjahr and Rauner, 2007 for an ACO approach to the nurse-scheduling problem) are not included in the revision. In addition, this paper concentrates only on uses of the ACO meta-heuristic on its own: Any "hybrid approach" is disregarded (Tkindt et al., 2002).

This paper contribution is twofold. In the first instance, it aims to help researchers apply this technique in production scheduling, demonstrating how research is being carried out in the literature. Therefore, some guidelines relating to the characteristics of the ACO algorithm applied to scheduling problems are derived. In the second instance, certain directions for future research in the field are highlighted.

To present the results, this paper is divided into six sections: Section 2 deals with the basics of scheduling problems; Section 3 is concerned with the ACO; Section 4 has to do with the classification method proposed and the papers reviewed; Section 5 presents an overview of the ACO application to scheduling problems; Section 6 shows a quantitative analysis of the literature; and Section 7 presents the final remarks.

LITERATURE REVIEW

Bauer et al., designed an algorithm for both approaches (1J Ti and 1J Pwi Ti), using well-known dispatch rules such as Earliest Due Date (EDD) to determine the visibility values. The authors used the Adjacent Pairwise Interchange (API) and a 2-opt heuristic as a local search.

den Besten et al., approached the problem 1J Pwi Ti. Their results showed that the ACO algorithm was able to obtain the best solution for all test instances. The local search algorithms used were an insert-interchange and an interchange-insert.

The papers by Merkle and Middendorf and Merkle and Middendorf proposed improvements to the work of Bauer et al., relating to the problem (wi.Ti). The authors developed a new visibility rule that constrained a deterministic criterion to restrict the search space and avoid generating poor solutions.

Also regarding the problem (wi.Ti), Holthaus and Rajendran used the Fast Ant Colony Optimization (FACO), which initializes the pheromone levels using a constructive heuristic. In this case, the problem characteristics were embedded in the pheromone matrix, allowing the visibility rule to have a constant value of $Z\%$

In this case, because the visibility rule did not need be calculated, the algorithm is faster to execute.

Cheng et al., implemented a hybrid algorithm for the (PTi) problem, based on results presented by Bauer et al. According to Cheng et al., the algorithm proposed contained a set of elimination rules that, inserted into the algorithm proposed by Bauer et al., simplified the search space and

improved the quality of the final solution.

The algorithm presented by Gagn et al., addressed a different problem, $1=sij=PTi$ (minimize total tardiness in a single machine environment with setup-dependent times). This algorithm reduced the search space using a candidate list. The visibility rule consisted of information related to

- The setup time of the job and
- The slack of the job. The transition rule, which the authors named as look-ahead information, used the visibility values and added information regarding the others non-processed tasks.

A similar problem, $1=sij=Pwi Ti$, was addressed by Liao and Juan. In this approach, which used the apparent tardiness cost with setup rule to calculate the value of the visibility, the authors used the job-to-position approach. It uses a graph that makes it possible to verify the likelihood of a job belonging to a position in the final sequence.

Kashan and Karimi approached the problem $1=batch$, incompatible, $sij=Pwi Ti$ (minimize total weighted tardiness on a single machine environment with formation of processing batches where some tasks cannot be processed in the same batch as others with sequence-dependent setup times). In this paper, the authors proposed two ACO algorithms, each one having different visibility functions.

Raghavan and Venkataramana approached the problem $Pm=Batch$, incompatible= $Pwi.Ti$. This problem has to do with minimizing the sum of weighted tardiness in a parallel machine environment that allows the formation of batches. The proposed algorithm consists of two phases:

- Generation of allowed batches using the dispatch rule Apparent Tardiness Cost (ATC) and
- An ACO based algorithm to schedule those batches on different machines.

Shyu et al., used the ACS algorithm to solve the $F2=nwt$, $setup=M$ problem (minimize makespan in a 2-machine non-per-mutational flowshop with sequence-dependent setups, in an environment that does not allow a wait between operations). In this algorithm, the visibility was defined using the job's setup and processing times.

DISCUSSION

ACO is one of the algorithms used for the discrete optimization problem, having been applied to solve many optimization problems in various domains. For single machine scheduling problems, we can cite the works of Liao and Juan. This algorithm is inspired by the searching behavior of ants for food, which can be described as follows: A group of ants starts searching in several random directions from the nest (this process is initiated only once). During the passage on any path, the ants produce a chemical pheromone to mark their paths. When an ant finds a source of food, it takes a quantity of it and returns to the nest by choosing the path with the largest amount of pheromone. The ant will then

start searching from the nest again, choosing the path that contains the largest amount of pheromone. The amount of pheromone is updated every time period. The shortest path will always contain the largest amount of pheromone and

therefore all the ants will move along it. To optimize the problem under study in this paper we adopted the ACO algorithm which was used (Table 1).

Table 1. Comparative analysis.

S. n o	Study of	Proposed method	Merits
1	Bauer et al.	1J Ti and 1J Pwi Ti	Earliest due date
2	den Besten et al.	insert– interchange and an interchange–insert.	Best solution for all test instances
3	Merkle and Middendorf	improvements to the work of Bauer et al.	Restrict search space and avoid generating poor solutions.
4	Holthaus and Rajendran	FACO	Faster to execute
5	Cheng et al.	hybrid algorithm for the (PTi)	simplified the search space and improved the quality of the final solution.
6	Gagn et al.	minimize total tardiness in a single machine environment with setup dependent times	Reduce search space using a candidate list
7	Liao and Juan	Addresses, 1=sij=Pwi Ti	graph that verify likelihood of a job belonging to a position in the final sequence.

CONCLUSION

This paper presents a literature review concerning the use of the ACO metaheuristic in scheduling problems. The scope of this survey covers four manufacturing environments: Single machine, parallel machine, flow shop, and job shop. Through the analysis, it was possible to verify that ant-based algorithms (ACO, ACS, and MMAS) are valid strategies for solving scheduling problems. Although the ACO algorithm is straightforward, the possibility to easily incorporate scheduling-specific heuristics and other metaheuristics increases the complexity and the research opportunities involving the use of this metaheuristic for this category of problems.

The analysis presented in this paper indicates that this field of research is a relatively new one, and only the initial results have been published. One can easily note that the first studies had to do with single machine scheduling problems. Only later were flow shop, parallel machine, and job shop environments addressed. The evolution of knowledge in this research area can also be observed in terms of the new features added to address certain problems.

REFERENCES

1. Zhuo X, Zhang J, Chen W (2007). A new pheromone design in ACS for solving JSP. IEEE Congr Evol Comput. 1963-1969.
2. Zhou P, Qingshan D (2009). Hybridizing fast taboo search with ant colony optimization algorithm for solving large scale permutation flow shop scheduling problem. IEEE Int Conf Granul Comput. 809-813.
3. Zhou H, Li Z, Wu X (2007). Scheduling unrelated parallel machine to minimize total weighted tardiness using ant colony optimization. IEEE Int Conf Granul Comput. 132-136.
4. Zhang Z, Zhang J, Li S (2010). A modified ant colony algorithm for the job shop scheduling problem to minimize makespan. Int Conf Mech Autom Cont Eng. 3627-3630.
5. Yoshikawa M, Terai H (2006). A hybrid ant colony optimization technique for job shop scheduling problems. Proc 4th Int Conf Soft Eng Res Manag. 95-100.
6. Ying KC, Liao CJ (2003). An ant colony system approach for scheduling problems. Prod Plann Cont. 14: 68–75.
7. Ying K, Lin S (2007). Multi-heuristic desirability ant colony system heuristic for non-permutation flowshop scheduling problems. Int J Adv Manuf Technol. 33: 793–802.
8. Yagmahan B, Yenisey MM (2008). Ant colony optimization for multi-objective flow shop scheduling problem. Comput Ind Eng. 54: 411–420.
9. Xu R, Chen H, Li X (2012). Makespan minimization on single batch-processing machine via ant colony optimization. Comput Oper Res. 39: 582-593.
10. Tkindt V, Monmarche N, Tercinet F, Laugt D (2002). An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. Eur J Oper Res. 142: 250–257.