



A Review of Implementing Ant System Algorithms on Scheduling Problems

Samar Kashef , Raafat Elshaer *

Zagazig University, Faculty of Engineering, Industrial Engineering Department, Zagazig, Egypt

ARTICLE INFO

Keywords:

Ant Systems,
Ant colony,
Optimization, Metaheuristics,
Machine shop scheduling.

ABSTRACT

The ant system (AS) and scheduling problem are well-known concepts in literature. Ant algorithms have been known to be an effective tool for solving combinatorial optimization problems. Elitist AS (EAS), rank-based AS (RAS), ant colony system (ACS), and max-min AS (MMAS) are the variants of the AS algorithm; they are triggered by the different ways of updating the pheromone trail τ , computing the visibility η , and/or other parameters in the basic AS model. The main contribution of this article is twofold. First, the basic AS and its controlled parameters are presented, the key variants of the ant algorithms are explained, and major changes of each variant from the basic model are tracked. Second, sixty papers are collected between 2015 and 2020 based on a search strategy for tracking the implementation of different AS variants in solving scheduling problems. Numerous findings based on a statistical analysis of the collected papers are reported and discussed. This study will allow the researcher to understand the essence of the ant algorithm, recognize the fundamental differences in its five systems, and determine how each of them can be implemented. Tracking a sample of articles that apply an ant algorithm for a specific case study gives researchers new ideas on how to adjust the original model to fit their problem.

1. Introduction

While the first problem, the traveling salesman (TSP), was addressed and solved using the original ant system (AS), other versions of AS were revised to be suitable for different combinatorial problems, such as supply chain designing, project planning, clustering, resource allocation, apps for the internet of things (IoT), scheduling, production planning, and process planning. As a result of the modern industrial revolution, scheduling issues have become more complex. Hence, researchers are directed to develop various intelligent systems in order to deal with the scheduling problems; various tools and techniques from artificial intelligence (AI) are used for the process. Article [1] reviewed a range of AI-based solution methods for different objective functions of the job shop scheduling problem. Article [2] mentioned five major branches of intelligent scheduling techniques, including machine learning, stochastic local search optimization algorithms (SLS), fuzzy logic, expert systems, and

constraint programming. SLS is classified into two types: neighborhood-based methods, such as simulated annealing, tabu search, and population-based methods, such as evolutionary approaches, ant colony optimization, and practical swarm optimization. All these metaheuristics were mentioned in a detailed classification tree by Elshaer and Awad [3], through studying the vehicle routing problem. [4] was the first to construct a curved model, to study ant behavior in nature. Then in the early nineties, [5] proposed an AS model as a heuristic approach for solving the optimization problem. In their paper, authors presented three AS variants: ant quantity, ant density, and ant cycle. To overcome the local optima and convergence issues in solving the TSP. After that, [6] proposed an AS model with a 3-opt local search. Later, [7] added the term “colony” to the AS algorithm as a common framework for all versions of the AS, which is now known as ant colony optimization (ACO). Actually, ACO was not the latest AS version but was still the best known at that time. One year later, [8]

* Corresponding author. Tel.: +02-01009137001
E-mail address: rheshaer@eng.zu.edu.eg

suggested a max-min ant model to improve the algorithm's efficiency. Researchers have not stopped developing ant algorithms that can be suitable for all optimization problems. [9] clarified the differences in pheromone trail management for the AS algorithms; basic AS, EAS, RAS, MMAS, and ACS by studying their performance in solving the car-sequencing problem as a test case. For a recent detailed study of the aforementioned five models of AS

algorithms we refer to [10]. Through the literature in the selected time of our survey, as mentioned in Section 4. We noticed that most of the review article focuses on reviewing scheduling problems or AS variants separately. Only a few review articles focus on solving a scheduling problem by using the ant method in general. Even in the article [11] the scheduling issue did not inserted, although the survey was about using the ant colony algorithm for solving some NP-Hard problems.

This paper has been arranged as follows. Section 2 explains the mechanism of an AS and the basic model. Section 3 discusses the extensions and variants of the AS algorithm. In Section 4, the search methodology for collecting relevant papers is mentioned. Section 5 is devoted to discussing the selected papers. Additionally, a sample of studies that have made changes to the equations of the original models has been mentioned. The paper is concluded in Section 6.

2. Basic Ant System (AS)

AS is an optimization algorithm inspired by the natural behavior of real ants. The main idea when using AS to solve problems with optimization depends on the motion of ants in a weighted graph (N, E) , where N is a set of nodes and E is a set of edges between nodes. The graph component values are modified by the ants through iterations. First, ants randomly choose the first node i , then to move from node i to node j , they follow a probability function represented by equation (1). When an ant k travels from node i to node j , a substance is placed on the edge between i and j , called trail (pheromone in real ants). The probabilistic decision depends on two parameters: trail intensity τ_{ij} , which represents information about how many ants in the past have chosen that same edge (i, j) , and trail visibility $\eta_{ij} = \frac{1}{d_{ij}}$, which says that the closer the node, the more desirable it is [5].

$$p_{ij} = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{h \in s_i^k} [(\tau_{ih})^\alpha (\eta_{ih})^\beta]} & \text{if } j \in s_i^k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where α and β indicate the weight of τ and η , respectively, d_{ij} is the heuristic distance between i and j , and s_i^k is the set of allowed nodes for ant k from node i . When an ant moves to the next candidate node, the pheromone trail $\tau_{ij}(t + 1)$ is updated according to equation (2), where $\tau_{ij}(t)$ is the current pheromone level and ρ is a constant, representing the trail evaporation coefficient. $\Delta\tau_{ij}(t, t + 1)$ is the pheromone laid on the edge (i, j) by all ants between time t and $t + 1$, calculated by equation (3), where $\Delta\tau_{ij}^k$ is the pheromone quantity deposit by ant k on edge (i, j) and m is the number of ants. Note that in the equations discussed later, we will not mention the term t and $t + 1$, it will be in the context. The initial pheromone value τ_0 usually equals zero or a given small value at the start of the algorithm.

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

$$\Delta\tau_{ij}(t, t + 1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t + 1) \quad (3)$$

Different choices about how to estimate $\Delta\tau_{ij}^k$ and when to update the τ_{ij} produce different models of the ant algorithm. If an ant k could detect the pheromone laid by the prior one on edge (i, j) , there is a high probability that this ant will follow the edge (i, j) . Therefore, the pheromone may be concentrated on the shortest track because it would be chosen by other ants; otherwise, $\Delta\tau_{ij}^k$ is set to be zero if ant k does not go from node i to node j . The pheromone trail may be updated by an ant k through three mechanisms, as mentioned in [5]. The first and the second mechanism perform the pheromone update as soon as the ant chooses the next step j ; this refers to ant density model equation (4) and ant quantity model equation (5). The third mechanism performs the pheromone update after determining a complete track by ant k at the end of each entire iteration; this refers to the ant cycle model equation (6), where Q is a constant, d_{ij} is the distance between (i, j) , and l_k is the total distance covered by ant $k \in \{1, \dots, m\}$.

$$\Delta\tau_{ij}^k = Q \quad (4)$$

$$\Delta\tau_{ij}^k = \frac{Q}{d_{ij}} \quad (5)$$

$$\Delta\tau_{ij}^k = \frac{Q}{l_k} \quad (6)$$

3. Extensions of the ant system algorithm

Several versions of the AS have been suggested in the literature to improve the efficiency of solving optimization problems, such as an EAS, RAS, ant colony system, and MMAS. For each of

them, we will include the mathematical model in the following subsections.

3.1. Elitist Ant System (EAS)

To improve the quality of the solution of AS, authors [12] suggested applying the elitist strategy with the AS algorithm. They added the quantity $\sigma \frac{Q}{L^*}$ to the pheromone trail of each edge of the best path found so far. The number of elitist ants, σ , is a parameter whose value is adjusted by tuning through applying the algorithm. To calculate the quantity of pheromone at the edge (i, j) for the EAS, authors [13] updated the equation (7) from the equation (2). Where $\Delta\tau_{ij}^*$ is the increase of trail level on edge (i, j) caused by an elitist ant, and it was calculated using equation (8). The number of elitist ants represented by $\sigma \leq m$, L^* is the tour length of the best solution found.

$$\tau_{ij} = \rho\tau_{ij} + \Delta\tau_{ij} + \Delta\tau_{ij}^* \quad (7)$$

$$\Delta\tau_{ij}^* = \begin{cases} \sigma \frac{Q}{L^*} & \text{if edge } (i, j) \text{ part of the best solution} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

3.2. Ranking Ant System (RAS)

R. F. Hartl [13] Suggested a rank-based algorithm. After all m ants had completed their tour, they were sorted based on their tour length ($l_1 \leq l_2 \leq \dots \leq l_\mu \leq \dots \leq l_m$). The contribution of an ant to the trail level is weighted according to the rank μ of the ant. The ranking was intended to solve the problem of ants' high pressure on the most selected edges. Pheromone is updated as in EAS model according to equation (7), where $\Delta\tau_{ij}$ is calculated by equation (9), $\Delta\tau_{ij}^\mu$ is the increase of trail level on edge (i, j) caused by the μ^{th} ant, and it is calculated by equation (10).

$$\Delta\tau_{ij} = \sum_{\mu=1}^{\sigma-1} \Delta\tau_{ij}^\mu \quad (9)$$

$$\Delta\tau_{ij}^\mu = \begin{cases} (\sigma - \mu) \frac{Q}{l_\mu} & \text{if best ant } \mu^{th} \text{ goes from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

3.3. Ant Colony System (ACS)

To improve AS performance, ACS was suggested by [6] by adding three main aspects to the

basic AS model that mentioned previously in Section 2. These three aspects explained in the following sub-sections.

3.3.1. State Transition Rule

In the basic AS, when an ant moves from node i to node j , it follows a random-proportional rule, as shown. in equation (1). But in ACS model, ants follow a pseudo-random-proportional rule as shown in equation (11), to select the index of the next node j , where $q \in [0,1]$ is a random number uniformly distributed and q_0 is a parameter that determines the percentage of exploitation and exploration for generating new solutions; if $q_0 = 1$, it is totally exploitation, and if $q_0 = 0$, it is totally exploration. J is a random variable representing the next node selected according to equation (1).

$$j = \begin{cases} \arg \max_{a \in s_i^k} \{(\tau_{ia})(\eta_{ia})^\beta\} & \text{if } q \leq q_0 \\ J & \text{otherwise} \end{cases} \quad (11)$$

3.3.2. Global Updating Rule

After all ants have completed their tour, the pheromone is updated according to equation (12), only by ants which constructed the shortest tour, where $\rho \in [0,1]$ is the global pheromone evaporation parameter and $\Delta\tau_{ij}$ is calculated according to equation (8) with $\sigma = Q = 1$.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho \Delta\tau_{ij} \quad (12)$$

3.3.3. Local Pheromone Updating Rule

In an advanced stage, when running the algorithm, ants would search in a narrow neighborhood of the best previous tour so they may give the same solution this problem called stagnation behavior. To solve this problem, the local pheromone updated strategy is applied by equation (13), where the desirability of the edges dynamically changes (i.e., every time an ant uses an edge, it loses some of its pheromones and becomes less desirable). To calculate $\Delta\tau_{ij}$, [6] suggested three experimental formulas: (14.1), (14.2), and (14.3), where $\epsilon \in [0,1]$ is the local pheromone evaporation parameter and $\gamma \in [0,1]$ is a parameter.

$$\tau_{ij} = (1 - \epsilon)\tau_{ij} + \epsilon \Delta\tau_{ij} \quad (13)$$

$$\Delta\tau_{ij} = \begin{cases} \gamma \cdot \max_{a \in s_i^k} \{\tau_{ja}\} & \text{where } 0 < \gamma < 1 \\ \tau_0 & \text{initial pheromone value} \\ 0 & \gamma = 0 \end{cases} \quad (14.1)$$

$$\tau_0 \quad (14.2)$$

$$\gamma = 0 \quad (14.3)$$

3.4. Max-Min Ant System (MMAS)

To avoid early search stagnation, the MMAS model was presented by [8]. The MMAS model differs from AS; it provides stronger exploitation of the global best solution by setting boundaries for the pheromone intensity, τ_{max} and τ_{min} , and initializing the pheromone trails to τ_{max} . The pheromone value is calculated and adjusted by equation (15), where $\Delta\tau_{ij}^*$ is calculated according to equation (8), with $\sigma = Q = 1$. If τ_{ij} is found to surpass τ_{max} , it is set back to τ_{max} , and if it is found to go below τ_{min} , it is reset to τ_{min} , and only one single ant (best ant) is used at the end of each iteration to refresh the pheromone.

$$\tau_{ij} = \max(\tau_{min}, \min(\tau_{max}, \rho\tau_{ij} + \Delta\tau_{ij}^*)) \quad (15)$$

By running the algorithm, these limits are updated by the time according to equations (16) and (17), where L^* could be iteration best or global best solution. The global best solution is developed with a probability of p_{best} ; n is the number of decision points, and $n/2$ represents the average number of edges at each decision point.

$$\tau_{max} = \frac{1}{1 - \rho} * \frac{1}{L^*} \quad (16)$$

$$\tau_{min} = \frac{\tau_{max} (1 - \sqrt[n]{p_{best}})}{(\frac{n}{2} - 1) \sqrt[n]{p_{best}}} \quad (17)$$

4. Search methodology

For tracking the implementation of different AS variants in solving scheduling problems, we conducted a literature review. We considered only relevant articles published between 2015 and 2020 in English-language journals and excluded others that were published in books, conference proceedings, and dissertations. In our survey we accessed the most trusted databases as, Elsevier, Springer, Emerald, IEEE, JSTOR, Taylor and Francis, and Wiley. Using Keywords: Ant Systems, Ant colony, Optimization, Metaheuristics, Machine shop scheduling. By applying the following search technique, the most important literature was extracted, and the strong relevant papers were retained. First, only articles containing the words “ant system algorithms” and “scheduling problem” in the title were selected. Second, the search was limited to articles published in journals with an impact factor of at least 1, based on the Impact Factors of 2019 by Thomson Reuters (2019). Third, the search was carried out in the domains of computers and industrial engineering, operations research and management science, transportation science, and technology.

5. Review results

AS a result of the search strategy mentioned in Section 4. The most important literature was extracted, and it was about 150 articles. Then, number of strong relevant papers were retained and resulted in final collection of sixty papers containing the most recent articles on the scheduling problem solved by the AS algorithm. Although the set was not exhaustive, it is considered a representative for the field. The distribution of articles per year during the survey period is illustrated in Fig.1 It is clear that the number of publications has grown in recent years. Table 1 summarizes the number of papers per journal and its impact factor, ordered alphabetically. It is clear that most of the papers surveyed were published by Computers & Industrial Engineering and IEEE Access.

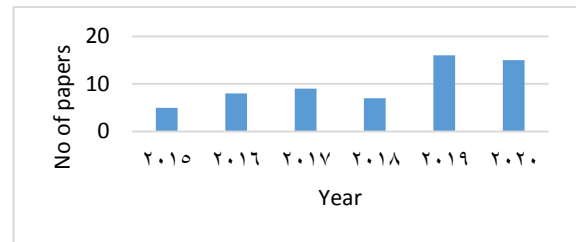


Fig. 1. Distribution of articles per year during the survey period

Table 1. Overview of the number of selected articles per journal and 2019 Impact Factor.

Journal	IF	# Papers
Alexandria engineering journal	2.46	1
Applied intelligence	3.325	2
Applied soft computing	5.472	4
Cluster computing	3.458	1
Computers & industrial engineering	4.135	13
Computers and operations research	3.424	2
Environmental modelling & software	4.807	1
European journal of operational research	4.123	1
Expert systems with applications	5.452	2
Flexible services and manufacturing journal	2.368	1
Future generation computer systems	6.125	3
IEEE transactions on cybernetics	11.079	1
IEEE access	3.745	8
IEEE transactions on industrial informatics	9.112	1
IEEE transactions on systems, man, and cybernetics: systems	9.309	1
Journal of ambient intelligence and humanized computing	4.594	1
Journal of cleaner production	7.246	1
Journal of intelligent manufacturing	4.311	5

Journal of scheduling	1.765	1
Knowledge-based systems	5.921	1
Neural computing and applications	4.774	1
Procedia computer science	1	1
Soft computing	3.050	1
Swarm and evolutionary computation	6.912	2
The international journal of advanced manufacturing technology	2.633	3
Transactions on intelligent transportation systems	6.319	1

A variety of scheduling problems are found in the literature. The sixty papers selected are summarized in Table 2. For each article, the table shows the year of publication, problem category, used ant model, and citation. The data in the table was ordered in descending order of the article publication year.

Table 2. Summarizing the literature review.

Year	Problem Category	Ant model	Citation
2020	Job Shop	ACS	[14]
2020	Scheduling of distributed cyber-physical system	ACS	[15]
2020	Health care issues	ACS	[16]
2020	Urban project planning	AS	[17]
2020	Resource allocation	AS	[18]
2020	Batch Scheduling	MMAS	[19]
2020	Resource allocation	ACS	[20]
2020	Resource allocation	ACS	[21]
2020	Batch Scheduling	ACS	[22]
2020	Job Shop	ACS	[23]
2020	Job Shop	ACS	[24]
2020	Time-Triggered flows in time-sensitive network	ACS	[25]
2020	Supply chain	ACS	[26]
2020	Scheduling the project critical chain	ACS	[27]
2020	Hot strip mill optimization	ACS	[28]
2019	Job Shop	ACS	[29]
2019	Flow Shop	ACS	[30]
2019	Job Shop	MMAS	[31]
2019	Batch Scheduling	ACS	[32]
2019	Workflow optimization problem	ACS	[33]
2019	Hybrid manufacturing	ACS	[34]
2019	Max-Mean dispersion optimization	AS	[35]
2019	Single-machine scheduling	ACS	[36]
2019	Assignment optimization	ACS	[37]
2019	Flow shop	ACS	[38]

2019	Virtual machine optimization	AS	[39]
2019	Cloud-fog task scheduling	ACS	[40]
2019	Process planning and scheduling	ACS	[41]
2019	Job Shop	MMAS	[42]
2019	Health care issues	ACS	[43]
2019	Batch Scheduling	ACS	[44]
2018	Flow Shop	ACS	[45]
2018	Job Shop	ACS	[46]
2018	Job Shop	ACS	[47]
2018	Assembly sequence planning	ACS	[48]
2018	Flow Shop	MMAS	[49]
2018	Resource allocation	MMAS	[50]
2018	Constraint satisfaction	ACS	[51]
2017	Job Shop	ACS	[52]
2017	Resource allocation	ACS	[53]
2017	Assembly sequence planning	ACS	[54]
2017	Flow Shop	AS	[55]
2017	Job Shop	ACS	[56]
2017	Finance-Based scheduling Problem	MMAS	[57]
2017	Load balanced scheduling of a grid processing system	ACS	[58]
2017	Computing systems workflow	ACS	[59]
2017	Irrigation scheduling	MMAS	[60]
2016	Batch Scheduling	MMAS	[61]
2016	Mobile application scheduling	MMAS	[62]
2016	Process planning and scheduling	AS	[63]
2016	Grid Scheduling	ACS	[64]
2016	Single-machine scheduling	ACS	[65]
2016	Cyclic scheduling	ACS	[66]
2016	Meta-scheduling	ACS	[67]
2016	Computing systems workflow	ACS	[68]
2015	Job Shop	ACS	[69]
2015	cellular manufacturing scheduling	MMAS	[70]
2015	Health care issues	ACS	[71]
2015	Process planning and scheduling	MMAS	[72]
2015	Parallel machine scheduling	MMAS	[73]

The distribution of papers on the ant model used is illustrated in Fig. 2 within the duration of the survey and under our restriction for this study. We found that 70% of the collected articles used ACS, 20% used MMAS, and only 10% used AS; no articles used the EAS or RAS model as the primary

algorithm to solve a scheduling problem. In addition, when reviewing the articles, we noticed that the title of some papers stated that the author would be using the AS model, but we found that they used ACS or MMAS, and vice versa.

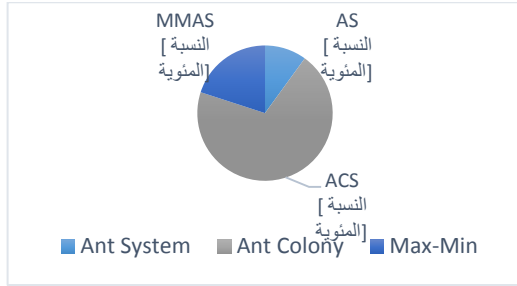


Fig. 2. Distribution of the sixty papers on the ant model used.

Through our study, we found that researchers would need to make some changes in the original model structure to satisfy their particular objectives and problem constraints. We selected a sample of studies that made changes to the equations of the original models previously described. In the following subsections, this will be stated for the AS, ACS, and MMAS models.

5.1. Ant System, AS

The survey showed that six papers from the collected publications solved the scheduling problem through the basic AS model. Two papers were represented in this subsection as an example of those which made improvements in the basic AS model, mentioned above in Section 2.

A. Ragmani, et al. [39] suggested an algorithm to define the optimal configuration for virtual machine placement in such a way that it optimizes the use of cloud computing resources. In their study, two algorithms are used to assign virtual machines to servers. The first algorithm concerns a dynamic scheduling policy. The second algorithm is a load-balancing technique that uses a fuzzy logic framework and the AS algorithm. Authors changed the basic AS probability function to be suitable for their problem, as shown in equation (18), where $\varphi_{ij}=1/(\text{total cloudlets})$, total cloudlets is the waiting queue request of a given virtual machine and u is the control factor of the queue of user requests.

$$p_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta (\varphi_{ij})^u}{\sum_{h \in s_i^k} [(\tau_{ih})^\alpha (\eta_{ih})^\beta (\varphi_{ij})^u]} \quad (18)$$

T. Cura and M. Özdemir, [35] addressed a parallel AS approach to solve the max-mean dispersion problem (MMDP). This problem refers to maximizing the mean dispersion between elements in a subset X , which was selected from Y . In the proposed model, an ant has to make one of two probability decisions: whether to insert or remove an element from the set. Parallel AS denotes that, ants perform a parallel search process. The MMDP problem differs in two points from the original problem (TSP): (1) the MMDP is a maximization problem while the TSP is a minimization problem and (2) the sequence of the selected elements is not important for the MMDP. As a result, the authors suggest updating visibility values as shown in equation (19), where $\gamma > 0$ is a parameter that regulates the effect of the latest best global solution, X_{gb} is the new global best solution found, and $f(X_{gb})$ is the fitness of the best solution. Note that visibility is constant in the original AS and is not modified during the search process.

$$\eta_i = \begin{cases} \eta_i + \gamma f(X_{gb}) & i \in X_{gb} \\ \eta_i & \text{otherwise} \end{cases} \quad (19)$$

5.2. Ant Colony System, ACS

The ant colony variant was considered the most known model of the AS. The reason could be because its mechanism is flexible and depends on the state transition rule, which gives more accuracy for the solution. As shown in Fig. 2, 42 papers used ACS. In this subsection, we presented some articles as an example of who has made changes in the basic ACS mentioned above in Subsection 3.3.

Job shop scheduling and conflict-free routing problem for automated guided vehicles (AGV) was solved earlier by using ACS. [69] described the mathematical model for this problem and modified the algorithm to meet its constraints. The problem was represented as a number of warehouse jobs to be processed on machines and a number of AGVs at the starting point of the warehouse, at the beginning of the planning period. Each AGV is assigned only one job: to move it from the starting point of the warehouse to the first pickup and delivery (P/D) point of the required machine. The AGV that is located on point $i \in I$ can be in one of the adjacent points $j \in s_i^k$ or the same point I , where I represents the set of coordinates of points on the network. The heuristic information η_{ij} , when AGV moves from one state to another, is calculated according to equation (20), where θ is a random positive number greater than one.

$$\eta_{ij} = \begin{cases} \theta & \text{if state } j \text{ is toward the goal} \\ 1 & \text{otherwise} \end{cases} \quad (20)$$

R. M’Hallah, et al. [65] proposed a hybrid ant colony heuristic model based on three techniques simulated annealing (SA), variable neighborhood search (VNS), and pairwise exchange for solving the total weighted earliness and tardiness single machine scheduling problems. In their model, the probability of assigning job j to position i is calculated as per equation (1). The heuristic information, η_{ij} , is calculated using a modified formula, as in equation (21), where \underline{w}_j and \overline{w}_j are the weighted earliness and tardiness of job j , respectively. The term $(\underline{w}_j E_j + \overline{w}_j T_j)$ represents the cost of assigning job j to position i , and $\max_{c \in s_i^k} \{ \underline{w}_c E_c + \overline{w}_c T_c \}$ represents the highest possible cost of filling position i among all alternative assignments of non-scheduled jobs, s_i^k .

$$\eta_{ij} = 1 - \frac{\underline{w}_j E_j + \overline{w}_j T_j}{\max_{c \in s_i^k} \{ \underline{w}_c E_c + \overline{w}_c T_c \}} \quad (21)$$

C. Ha [41] suggested a modified ant colony heuristic model for solving the integrated process planning and scheduling problem (IPPS). The high flexibility and complexity of IPPS need a broad archive size for pheromone trail, which needs lots of computation time to test them. The author had proposed an approach that increases the pheromone by a constant incremental amount $\Delta\tau$ without allowing evaporation, as in equation (22). Author suggested that this updating feature saves a lot of computational energy because it only updates a few edges that belong to one solution path rather than the entire pheromone trail.

$$\tau_{ij} = \max\{\tau_{\max}, \tau_{ij} + \Delta\tau\} \quad (22)$$

S. Zhang, et al. [14] addressed the assembly scheduling problem for a flexible job shop environment with two adjacent working areas, whose products are integrated with flexible non-linear process plans and assembling operations. The assembling structures of products are either flat or multi-levelled. In this problem, there are three objective functions. Authors have solved their model using a distributed ant colony approach mixed with MMAS. In their research, the distributed model consisted of seven colonies. The first three colonies (1, 2, 3) were assigned one for each objective. The second three colonies (4, 5, 6) were assigned one for each pair of objectives. The seventh colony was assigned for all three objectives. η represents the local optimality of alternative assignments and its values were computed in different ways according to the objective function it concerns, as in equation (23) for colonies (1, 2, 3). For an ant k from colony

4, 5, 6, or 7 assigned with more than one objective, η is calculated using equation (24).

$$\begin{cases} \eta_{1u} = \frac{1}{c_u^*} \\ \eta_{2u} = \frac{1}{c_u^* + d_{ju} + f_{ju}} \\ \eta_{3u} = \frac{1}{p_u} \end{cases} \quad u \in U \quad (23)$$

where U is a set of the assignment (assignment u is the association of an operation with an alternative machine and the alternative assignments are sorted by the η values in decreasing order). For an assignment u : job j_u , expected completion time c_u^* , processing time p_u , job j due date d_j , and objective function f_j .

$$\eta'_{iu} = \sum_{i=1}^3 w_{ik} \frac{\eta_{iu}}{\bar{\eta}_i} \quad (i = 4, 5, 6, 7) \quad (24)$$

where $\bar{\eta}_i$ is the average value of η_{iu} for all alternative assignments and w_{ik} is the weight of objective i for ant k .

E. Martin, et al. [16] proposed an ant colony model to solve the home care scheduling problem. This problem is to build a set of routes used by caregivers who provide daily care to patients located in a definite area at a specific time. Each cluster is a feasible weekly schedule for a single caregiver. The heuristic distance is represented by the unproductive time (waiting + traveling) for the caregiver between patient i and patient j . Each caregiver has a service track, consisting of a set of tasks (patients). Local pheromone update was performed according to equation (13), formula (14.2). The pheromone global update was performed according to equation (12). $\Delta\tau_{ij}$ is calculated according to equation (4), where Q in this case was not a constant and was defined by equation (25).

$$Q = \frac{p}{Tot} * \min\left(1, \frac{Tot}{C}\right) \quad (25)$$

where p is the service time, Tot is total service time (productive + unproductive), the term $\frac{Tot}{C}$ represents a penalty time for each hour, and C represents the attenuation of cluster quality. Note that the parameters are specific to this article’s proposed method.

5.3. Max-Min Ant System, MMAS

Our survey showed 12 articles that used MMAS in solving scheduling problems. To map the differences, we present examples of papers that changed the basic model of MMAS, mentioned

above in Section 3.4. [53] made some modifications to the model to solve the railway rescheduling problem as the MMAS has no built-in mechanism for addressing dynamic multi-objective issues. They designed four distinct algorithm forms to either preserve or delete the pheromone and the solution archive after changes occurred. The maximum and minimum boundaries of the pheromone level was determined, as in equations (26) and (27), where c is the fitness of the best ant and a is a constant parameter of the algorithm.

$$\tau_{max} = \frac{1}{L^*} \quad (26)$$

$$\tau_{min} = \frac{\tau_{max}}{a} \quad (27)$$

On the Internet of Manufacturing Things (IoMT) environment, many abnormal event disturbances occurred, such as machine breakdown, urgent order arrival, etc. To tackle the issue of real-time control in energy-efficient shop floor schedule, [42] proposed a metaheuristic model called the PN-ACS algorithm, which combines a representation tool based on timed transition petri nets (TTPN) and MMAS. η_{ij} , in this problem, represents the actual energy cost required and calculated, as in equation (28), where $ET(x)$ is the real response time for transition x between nodes (i, j) , $DT(x)$ is the time delay associated with the transition x , pe_M processing power of machine M , and ne_M is the no-load power of machine M .

$$\eta_{ij} = ne_M \cdot ET(x) + pe_M \cdot DT(x) \quad (28)$$

6. Conclusion

This paper provides a literature review on the use of AS metaheuristic models in solving scheduling problems. The scope of this survey includes papers published in the last five years with the main terms, "ant system algorithms and scheduling problem." The main contribution of this article is twofold: First, we collected the main types of ant system algorithms, which are AS, EAS, RAS, ant colony, and MMAS. The differences between them are monitored in the way of updating the algorithm parameters. Second, based on a proposed search strategy, a final group of sixty papers is collected and classified based on the used AS and the research problem category. The statistical analysis of the collected papers shows that about 70%, 20%, and 10% of papers used the ant colony algorithm, the max-min system, and the AS, respectively. Sometimes, authors hybridize the ant colony model with another type of metaheuristic model, such as genetic algorithm, local search, greedy technique, or with other variants of ant

models. Based on the statistical analysis, many findings, which provide researchers with new ideas about how to modify the original model to suit their problem, are reported and discussed.

In most of literature, we found that, machine shop scheduling problem not being solved by ant system algorithm, although it was proved to be effective technique for optimization problem. we suggest for future work developing algorithms based on ant system models for solving the different machine shop scheduling problem as, job shop, open shop, flow shop, ..., and mixed shop.

References

- [1] B. Çaliş and S. Bulkan, "A research survey: review of AI solution strategies of job shop scheduling problem," *J. Intell. Manuf.*, vol. 26, no. 5, pp. 961–973, Oct. 2015.
- [2] M. H. Fazel Zarandi, A. A. Sadat Asl, S. Sotudian, and O. Castillo, "A state of the art review of intelligent scheduling," *Artif. Intell. Rev.*, vol. 53, no. 1, pp. 501–593, Jan. 2020.
- [3] R. Elshaer and H. Awad, "A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants," *Comput. Ind. Eng.*, vol. 140, no. December 2019, p. 106242, Feb. 2020.
- [4] U. L. De Bruxelles, "Probabilistic Behaviour in Ants: A Strategy of Errors?," pp. 259–271, 1983.
- [5] M. Dorigo, V. Maniezzo, and a Colomi, "The ant system: An autocatalytic optimizing process," TR91-016, Politec. di Milano, pp. 1–21, 1991.
- [6] M. Dorigo, S. Member, and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," vol. 1, no. 1, pp. 53–66, 1997.
- [7] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," *Proc. 1999 Congr. Evol. Comput. CEC 1999*, vol. 2, pp. 1470–1477, 1999.
- [8] T. Stützle and H. H. Hoos, "MAX – MIN Ant System," vol. 16, pp. 889–914, 2000.
- [9] C. Gagné, M. Gravel, S. Morin, and W. L. Price, "Impact of the pheromone trail on the performance of ACO algorithms for solving the car-sequencing problem," *J. Oper. Res. Soc.*, vol. 59, no. 8, pp. 1077–1090, 2008.
- [10] A. Prakasam and N. Savarimuthu, "Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants," *Artif. Intell. Rev.*, vol. 45, no. 1, pp. 97–130, Jan. 2016.
- [11] A. K. Mandal and S. Dehuri, "A Survey on Ant Colony Optimization for Solving Some of the Selected NP-Hard Problem," pp. 85–100, 2020.
- [12] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 26, no. 1, pp. 29–41, 1996.
- [13] R. F. Hartl, "A New Rank Based Version of the Ant System - A Computational Study Bernd Bullnheimer," no. 1, 1997.
- [14] S. Zhang, X. Li, B. Zhang, and S. Wang, "Multi-objective optimisation in flexible assembly job shop scheduling using a distributed ant colony system," *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 441–460, 2020.
- [15] N. Yi, J. Xu, L. Yan, and L. Huang, "Task optimization and scheduling of distributed cyber-physical system based on improved ant colony algorithm," *Futur. Gener. Comput. Syst.*, vol. 109, pp. 134–148, 2020.
- [16] E. Martin, A. Cervantes, Y. Saez, and P. Isasi, "IACS-HCSP: Improved ant colony optimization for large-scale home care scheduling problems," *Expert Syst. Appl.*, vol. 142, 2020.
- [17] B. Khelifa and M. R. Laouar, "A holonic intelligent decision support system for urban project planning by ant colony optimization algorithm," *Appl. Soft Comput. J.*, vol. 96, p. 106621, 2020.
- [18] H. Wang, J. Li, J. Tian, and K. A. N. Wang, "WebIDE Cloud Server Resource Allocation With Task Pre-

- Scheduling in IoT Application Development,” IEEE Access, vol. 8, pp. 16216–16224, 2020.
- [19] H. Zhang, Z. hong Jia, and K. Li, “Ant colony optimization algorithm for total weighted completion time minimization on non-identical batch machines,” *Comput. Oper. Res.*, vol. 117, 2020.
- [20] Z. Y. Zhang and X. Zhang, “Shared Bikes Scheduling under Users’ Travel Uncertainty,” IEEE Access, vol. 8, pp. 3123–3143, 2020.
- [21] R. Chen, C. Liang, D. Gu, and H. Zhao, “A competence-time-quality scheduling model of multi-skilled staff for IT project portfolio,” *Comput. Ind. Eng.*, vol. 139, no. November 2019, p. 106183, 2020.
- [22] Z. Jia, L. Gao, and X. Zhang, “A new history-guided multi-objective evolutionary algorithm based on decomposition for batching scheduling,” *Expert Syst. Appl.*, vol. 141, p. 112920, 2020.
- [23] Y. Wan, T.-Y. ZUO, L. Chen, W.-C. Tang, and J. Chen, “Efficiency-Oriented Production Scheduling Scheme: An Ant Colony System Method,” IEEE Access, vol. 8, pp. 19286–19296, 2020.
- [24] W.-R. Jong, H.-T. Chen, Y.-H. Lin, Y.-W. Chen, and T.-C. Li, “The Multi-layered Job-shop Automatic Scheduling System of Mould Manufacturing for Industry 3.5,” *Comput. Ind. Eng.*, vol. 149, p. 106797, 2020.
- [25] Y. Wang et al., “A time-sensitive network scheduling algorithm based on improved ant colony optimization,” *Alexandria Eng. J.*, 2020.
- [26] I. Dzalbs and T. Kalganova, “Accelerating supply chains with Ant Colony Optimization across a range of hardware solutions,” *Comput. Ind. Eng.*, vol. 147, no. June, p. 106610, 2020.
- [27] X. Zhang and W. Liu, “Complex Equipment Remanufacturing Schedule Management Based on Multi-Layer Graphic Evaluation and Review Technique Network and Critical Chain Method,” IEEE Access, vol. 8, pp. 108972–108987, 2020.
- [28] R. Zhang, S. Song, S. Member, and C. Wu, “Local Search Enhanced Ant Colony,” vol. 16, no. 4, pp. 2809–2819, 2020.
- [29] I. Chaouch, O. B. Driss, and K. Ghedira, “A novel dynamic assignment rule for the distributed job shop scheduling problem using a hybrid ant-based algorithm,” *Appl. Intell.*, vol. 49, no. 5, pp. 1903–1924, 2019.
- [30] W. Qin, Z. Zhuang, Y. Liu, and O. Tang, “A two-stage ant colony algorithm for hybrid flow shop scheduling with lot sizing and calendar constraints in printed circuit board assembly,” *Comput. Ind. Eng.*, vol. 138, no. October, p. 106115, 2019.
- [31] R. Behmanesh and M. Zandieh, “Surgical case scheduling problem with fuzzy surgery time: An advanced bi-objective ant system approach,” *Knowledge-Based Syst.*, vol. 186, p. 104913, 2019.
- [32] Z. hong Jia, Y. Wang, C. Wu, Y. Yang, X. yi Zhang, and H. ping Chen, “Multi-objective energy-aware batch scheduling using ant colony optimization algorithm,” *Comput. Ind. Eng.*, vol. 131, no. November 2018, pp. 41–56, 2019.
- [33] Y. Gong, T. Gu, F. Zhao, and H. Yuan, “Multiobjective Cloud Workflow Scheduling: Approach,” *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, 2019.
- [34] W. Liu et al., “Production planning for stochastic manufacturing/remanufacturing system with demand substitution using a hybrid ant colony system algorithm,” *J. Clean. Prod.*, vol. 213, pp. 999–1010, 2019.
- [35] T. Cura and M. Özdemir, “Computers & Industrial Engineering A new use of the ant system algorithm for the max-mean dispersion problem,” *Comput. Ind. Eng.*, vol. 135, no. March, pp. 628–642, 2019.
- [36] S. Rubaiee and M. B. Yildirim, “An energy-aware multiobjective ant colony algorithm to minimize total completion time and energy cost on a single-machine preemptive scheduling,” *Comput. Ind. Eng.*, vol. 127, no. December 2018, pp. 240–252, 2019.
- [37] W. U. Deng, J. Xu, and H. Zhao, “An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem,” IEEE Access, vol. 7, pp. 20281–20292, 2019.
- [38] M. Kurdi, “Ant colony system with a novel Non-DaemonActions procedure for multiprocessor task scheduling in multistage hybrid flow shop,” *Swarm Evol. Comput.*, vol. 44, no. October 2018, pp. 987–1002, 2019.
- [39] A. Ragmani, A. Elomri, N. Abghour, K. Moussaid, and M. Rida, “FACO: a hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing,” *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 10, pp. 3975–3987, 2019.
- [40] J. Xu, Z. Hao, R. Zhang, and X. Sun, “A Method Based on the Combination of Laxity and Ant Colony System for Cloud-Fog Task Scheduling,” IEEE Access, vol. 7, pp. 116218–116226, 2019.
- [41] C. Ha, Evolving ant colony system for large - sized integrated process planning and scheduling problem considering sequence - dependent setup times, no. 0123456789. Springer US, 2019.
- [42] S. Tian, T. Wang, L. E. I. Zhang, and X. Wu, “An Energy-Efficient Scheduling Approach for Flexible Job Shop Problem in an Internet of Manufacturing Things Environment,” IEEE Access, vol. 7, pp. 62695–62704, 2019.
- [43] J. Decerle, O. Grunder, A. Hajjam El Hassani, and O. Barakat, “A hybrid memetic-ant colony optimization algorithm for the home health care problem with time window, synchronization and working time balancing,” *Swarm Evol. Comput.*, vol. 46, no. January, pp. 171–183, 2019.
- [44] Z. Jia, J. Yan, J. Y. T. Leung, K. Li, and H. Chen, “Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities,” *Appl. Soft Comput. J.*, vol. 75, pp. 548–561, 2019.
- [45] O. Engin, A. Güçlü, and A. Güc, “A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems,” *Appl. Soft Comput. J.*, vol. 72, pp. 166–176, 2018.
- [46] B. Zhao, J. Gao, K. Chen, and K. Guo, “Two-generation Pareto ant colony algorithm for multi-objective job shop scheduling problem with alternative process plans and unrelated parallel machines,” *J. Intell. Manuf.*, vol. 29, no. 1, pp. 93–108, 2018.
- [47] S. Zhang and T. N. Wong, “Integrated process planning and scheduling: an enhanced ant colony optimization heuristic with parameter tuning,” *J. Intell. Manuf.*, vol. 29, no. 3, pp. 585–601, 2018.
- [48] C. Y. Ng, “Green product design and development using life cycle assessment and ant colony optimization,” *Int. J. Adv. Manuf. Technol.*, vol. 95, no. 5–8, pp. 3101–3109, 2018.
- [49] W. Q. J. Z. D. Song, W. Qin, J. Zhang, and D. Song, “An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time,” *J. Intell. Manuf.*, vol. 29, no. 4, pp. 891–904, 2018.
- [50] Z. Zhang, F. Hu, and N. Zhang, “Ant colony algorithm for satellite control resource scheduling problem,” *Appl. Intell.*, vol. 48, no. 10, pp. 3295–3305, 2018.
- [51] Q. Zhang and C. Zhang, “An improved ant colony optimization algorithm with strengthened pheromone updating mechanism for constraint satisfaction problem,” *Neural Comput. Appl.*, vol. 30, no. 10, pp. 3209–3220, 2018.
- [52] I. Chaouch, O. B. Driss, and K. Ghedira, “ScienceDirect A Modified Modified Ant Ant Colony Colony Optimization Optimization algorithm algorithm for for the the Distributed Distributed Job Job shop shop Scheduling Scheduling Problem Problem,” *Procedia Comput. Sci.*, vol. 112, pp. 296–305, 2017.
- [53] J. Eaton, S. Yang, S. Member, and M. Gongora, “Ant Colony Optimization for Simulated Dynamic Multi-Objective Railway Junction Rescheduling,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 2980–2992, 2017.
- [54] D. Wang, X. Shao, and S. Liu, “Assembly sequence planning for reflector panels based on genetic algorithm and ant Colony optimization,” *Int. J. Adv. Manuf. Technol.*, vol. 91, no. 1–4, pp. 987–997, 2017.
- [55] C. Chamnanlor, K. Sethanan, M. Gen, and C. F. Chien, “Embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints,” *J. Intell. Manuf.*, vol. 28, no. 8, pp. 1915–1931, 2017.

- [56] F. El Khoukhi, J. Boukachour, and A. El Hilali Alaoui, "The 'Dual-Ants Colony': A novel hybrid approach for the flexible job shop scheduling problem with preventive maintenance," *Comput. Ind. Eng.*, vol. 106, pp. 236–255, 2017.
- [57] S. T. Al-Shihabi and M. M. AlDurgam, "A max-min ant system for the finance-based scheduling problem," *Comput. Ind. Eng.*, vol. 110, pp. 264–276, 2017.
- [58] D. P. Mahato, R. S. Singh, A. K. Tripathi, and A. K. Maurya, "On scheduling transactions in a grid processing system considering load through Ant Colony Optimization," *Appl. Soft Comput. J.*, vol. 61, pp. 875–891, 2017.
- [59] B. Xiang, B. Zhang, and L. Zhang, "Greedy-Ant: Ant Colony System-Inspired Workflow Scheduling for Heterogeneous Computing," *IEEE Access*, vol. 5, pp. 11404–11412, 2017.
- [60] D. C. H. Nguyen, J. C. Ascough, H. R. Maier, G. C. Dandy, and A. A. Andales, "Optimization of irrigation scheduling using ant colony algorithms and an advanced cropping system model," *Environ. Model. Softw.*, vol. 97, pp. 32–45, 2017.
- [61] N. Rafiee Parsa, B. Karimi, and S. M. Moattar Husseini, "Minimizing total flow time on a batch processing machine using a hybrid max–min ant system," *Comput. Ind. Eng.*, vol. 99, pp. 372–381, 2016.
- [62] X. Wei, J. Fan, T. Wang, and Q. Wang, "Efficient application scheduling in mobile cloud computing based on MAX–MIN ant system," *Soft Comput.*, vol. 20, no. 7, pp. 2611–2625, 2016.
- [63] X. Liu, Z. Ni, and X. Qiu, "Application of ant colony optimization algorithm in integrated process planning and scheduling," *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 1–4, pp. 393–404, 2016.
- [64] P. K. Tiwari and D. P. Vidyarthi, "Improved auto control ant colony optimization using lazy ant approach for grid scheduling problem," *Futur. Gener. Comput. Syst.*, vol. 60, pp. 78–89, 2016.
- [65] R. M'Hallah, A. Alhajraf, R. M. Hallah, and A. Alhajraf, "Ant colony systems for the single-machine total weighted earliness tardiness scheduling problem," *J. Sched.*, vol. 19, no. 2, pp. 191–205, 2016.
- [66] A. Elmi and S. Topaloglu, "Multi-degree cyclic flow shop robotic cell scheduling problem: Ant colony optimization," *Comput. Oper. Res.*, vol. 73, pp. 67–83, 2016.
- [67] G. S. Pavani and R. I. Tinini, "Distributed meta-scheduling in lambda grids by means of Ant Colony Optimization," *Futur. Gener. Comput. Syst.*, vol. 63, pp. 15–24, 2016.
- [68] S. Kianpisheh, N. M. Charkari, and M. Kargahi, "Ant colony based constrained workflow scheduling for heterogeneous computing systems," *Cluster Comput.*, vol. 19, no. 3, pp. 1053–1070, 2016.
- [69] M. Saidi-mehrabad, S. Dehnavi-arani, F. Evazabadian, and V. Mahmoodian, "An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs," *Comput. Ind. Eng.*, vol. 86, pp. 2–13, 2015.
- [70] D. Li, M. Li, X. Meng, and Y. Tian, "A hyperheuristic approach for intercell scheduling with single processing machines and batch processing machines," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 45, no. 2, pp. 315–325, 2015.
- [71] W. Xiang, J. Yin, and G. Lim, "An ant colony optimization approach for solving an operating room surgery scheduling problem," *Comput. Ind. Eng.*, vol. 85, pp. 335–345, 2015.
- [72] B. Y. Cheng, J. Y. T. Leung, and K. Li, "Integrated scheduling of production and distribution to minimize total cost using an improved ant colony optimization method," *Comput. Ind. Eng.*, vol. 83, pp. 217–225, 2015.
- [73] R. F. Tavares Neto, M. Godinho Filho, and F. M. da Silva, "An ant colony optimization approach for the parallel machine scheduling problem with outsourcing allowed," *J. Intell. Manuf.*, vol. 26, no. 3, pp. 527–538, 2015.