Economic Computation and Economic Cybernetics Studies and Research, Issue 4/2021; Vol. 55

Assistant Professor Reza BEHMANESH, PhD
E-mail: rezaehs@yahoo.com
Department of Industrial Engineering
Naghshejahan Higher Education Institute, Isfahan, Iran
Lecturer Iman RAHIMI, PhD
E-mail: iman1364@gmail.com
Young Researchers and Elite Club, Islamic Azad University
Isfahan, Iran

IMPROVED ANT COLONY OPTIMIZATION FOR MULTI-RESOURCE JOB SHOP SCHEDULING: A SPECIAL CASE OF TRANSPORTATION

Abstract. *In this paper, we addressed the multi-resource job shop scheduling problem (MRJSP) with resource flexibility to minimize makespan. Regarding this issue, a mixed integer linear programming (MILP) model was developed. The minimization of makespan is a non-deterministic polynomial time optimization problem; hence, we implemented several improvements to the ant colony optimization (ACO) algorithm. To improve the algorithm, a new strategy inspired from selfish herd (SH) theory was used to update the pheromones in the algorithm. Scheduling WOs in a transportation workshop is similar to a job shop scheduling problem. Therefore, the structure of this special case was chosen to evaluate the new method. The proposed approach was evaluated in five test cases and numerical computational experiments. The performance of the improved ACO algorithm was compared in all test cases, including small and large test cases. The computational experiments show that the algorithms modified with the SH strategy outperformed the same algorithms without this strategy. Moreover, the results demonstrate the efficiency and capability of the proposed model and solution approach for optimizing the makespan of the problem.*

Keywords*: Evolutionary computation; Multi-resource job shop scheduling problem; Ant system; Selfish herd; Makespan.*

JEL Classification: C61

1. Introduction

Production scheduling as a decision-making process is found not only in manufacturing industries but also in service systems (Pinedo, 2012). Meanwhile, efficient and optimal production schedules lead to key improvements in cost

DOI: 10.24818/18423264/55.4.21.18

reduction and productivity. Therefore, scheduling problems are very important in production systems not only for researchers but also for practitioners (Huang and Yang, 2008). Among the scheduling problems, job shop scheduling problem (JSP) is a common scheduling model that is applied in real-world industries. In JSP, it is assumed that only a machine or resource is assigned to process each job, while in real-world practices such as a company's transportation system, several resources such as human resources and vehicles must be allocated to each job that is performed. In other words, multiple resources should process jobs simultaneously.

In this paper, we were dealing with a multi-resource job shop scheduling problem in which all jobs were processed on multiple resources. The parameters, such as processing time, were assumed deterministic, and also resources were in excess. The contribution of the paper is in the field of operations research technique. A novel pheromone update strategy was introduced for the ant colony optimization (ACO) algorithm, and according to that, a new algorithm was proposed to schedule jobs in the job shop environment to minimize maximum completion time (makespan). In brief, following results are generated in this paper;

- extension of multi resources job shop scheduling (MRJSP) problem according to the transportation department of a manufacturing company as a case study.
- A new multiple constraint MILP formulation is developed in which multiple resources are addressed based on the structure of the transportation system.
- Several versions of the ACO algorithm are developed to solve the MRJSP in a transportation environment.
- An ACO algorithm was developed and improved by using a new update pheromone strategy called selfish heard (SH) for solving the problem.

The next sections are described as follows; In section 2, we gave a brief overview of the previous studies on the job shop scheduling regarding the scope of the problem. In section 3, problem and the special case of the transportation in a manufacturing company were stated, and then, in section 4; we proposed the ant colony optimization and other improvements to achieve the best algorithm among them. Section 5, provides illustrative examples, computational experiments, and lastly, we concluded and presented some suggestions for future research in section 6.

2. Background and related work

Each job ordered in a job shop environment must be scheduled based on the route of processing by available resources on the floor shop. This scheduling problem is positioned into an operational level of decision making problems and is divided in two sub-problems: (i) allocation, in which each operation is assigned to a resource and (ii) sequencing, where the sequence of operations is determined. The following literature review is organized in two subsections: application and methodology.

2.1. Application

Several studies in the literature extended the JSP and optimized several objective functions based on new parameters and constraints (Artigues et al, 2005) considered the JSP with sequence-dependent setup times (SDST). (Guo et al, 2006) studied the JSP and constructed a general mathematical model to minimize the total penalties of earliness and tardiness. In another study, (Huang and Yang, 2008) addressed the JSP with a time window to minimize earliness and tardiness as a total penalty. (Pan and Huang, 2009) modeled a no-wait JSP to minimize total completion time. (Mati, 2010) focused on minimizing makespan in a job-shop environment in which equipment was not accessible during the planning phase. (Liu and Kozan, 2016) focused the JSP in which buffering requirements were considered and then a mixed integer programming was proposed to solve the problem. In contrast, a few researchers extended the JSP using multiple resource in actual cases. For example, (Mati and Xie, 2008) addressed the MRJSP with resource flexibility (MJSPF) to minimize makespan. Also (Behmanesh et al, 2020) extended the FJSP in operating room. In this problem, each operation may require simultaneous processing by integrating resources.

To the best of our knowledge, a multi-constraint and multi-resource JSP with resource flexibility has not been studied in a transportation system. We believe this is the first time that constraints, parameters, and variables of a transportation process were applied to MRJSP modeling to optimize makespan. Thus, no previous studies were performed that modeled the mathematical programming of MRJSP for a transportation department.

2.2. Methodology

There are two types of methodologies described in the literature to handle the kinds of combinatorial scheduling problems, which include exact algorithms and meta-heuristic algorithms described in several studies (Beck et al, 2011; Grimes and Hebrard, 2015). The exact algorithms that are applied to solve scheduling problems are integer programming. However, these methods become inefficient with an increase in the numbers of jobs. As the size of the problem increases, the computational time to solve the problem increases as well. Thus, new approaches such as evolutionary algorithms are used to find near-optimal or approximation solutions in as short of a time as possible. Since scheduling problems (especially JSP) are categorized as nondeterministic polynomial time hardness (NP-hard) combinatorial problems (Kundakci and Kulak, 2016), evolutionary algorithms are suitable methods to solve scheduling problems. For instance, tabu search (Ferland et al, 2001) was used as a meta-heuristic algorithm. For solving the extension of JSP, a polynomial algorithm case with two jobs (Mati and Xie, 2008) was proposed as exact approaches. Furthermore, evolutionary computation approaches (metaheuristics) were proposed to solve the combinatorial nature of the MRJSP. In the field of job-shop problems, heuristic or meta-heuristic procedures were developed to achieve near-optimal solutions, such as genetic algorithm (Guo et al, 2006; Asadzadeh and Zamanifar, 2010), the hybrid evolutionary or genetic algorithms (Mati and Xie, 2008; Pan and Huang, 2009; Kundakci and Kulak, 2016; Liu and Kozan, 2016), the memetic algorithm (Salido et al, 2017), a tabu thresholding heuristic (Mati, 2010), and ACO (Huang and Yang, 2008). Several researchers considered several advanced ACO algorithms for various scheduling problems. (Tiwari and Vidyarthi, 2016) applied an improved auto control ACO to solve a grid scheduling problem. (Heinonen and Pettersson, 2007) employed a hybrid ACO for solving a JSP.

The structure of the ACO algorithm as a constructive algorithm is compatible with a MRJSP, due to the fact that a constructive algorithm always generates a feasible solution with an associated savings in time. In contrast, improvement algorithms may generate infeasible solutions for a MRJSP after using operators such as swap, crossover, insertion, etc. and hence more time may be needed to repair the infeasible solutions. Therefore, several versions of the ACO algorithm were chosen to solve the MRJSP in this study. To the best of our knowledge, no studies have been performed in which novel versions of the ACO algorithm were developed and used to solve the MRJSP in a transportation case. Accordingly, the solution techniques developed in this work for this problem are novel. The first ACO algorithm, the ant system (AS), was introduced by (Dorigo et al, 1991) and was developed. A new strategy termed SH for the pheromone updating was introduced into all the algorithms. Finally, all the proposed solutions were compared and the efficiency of the method was determined according to the quality of the solutions.

3. Problem Statement

In order to describe the problem statement section, the details have been divided in to two subsections; a) the structure of transportation system scheduling, b) a mathematical programming for transportation system scheduling as a special case of MRJSP.

3.1. Description of structure of transportation system scheduling

In this section, the processing orders associated with the transportation system of a manufacturing company are described. This section highlights the novelty of the proposed model. In this work, for the first time, a model is proposed that is based on the variables and the constraints of the MRJSP in the transportation system. We used scheduling similarities between the structure of the transportation system and the job shop environment.

Job-scheduling associated with a manufacturing company’s transportation system is the main subject addressed in this work. Other units of the company place demands on the transportation system, including work orders (WOs) for the transportation system. Orders (jobs) include maintenance and transportation tasks that are processed and assigned multiple resources by the transportation system. Other units of the company demand vehicles and experts from the transportation unit for processing WOs, such as maintenance and transportation. A unit may demand a job consisting of several operations (tasks), with each operation requiring multiple resources for processing. Each operation is processed once by a machine but several operations may be required for jobs operated by the same machine along with different human resources. It is assumed that all orders are deterministic and are accessible for scheduling (these orders were assumed to be part of the scope of the proposed problem). Units demand services from the transportation system to complete required maintenance and transportation. Thus, some departments of the company are considered to be customers of the transportation system. Three resource types are allocated to each WO including drivers, riggers, and machines (vehicles such as crane and trailers and various types of equipment). These resources should be assigned to a WO operation simultaneously and this system is considered to be a multi-resource assigning system. Due to the shortage of resources, all resources are not available for WO-allocation simultaneously. Therefore, the completion time of all demands is increased. Planning and scheduling play crucial role in transportation management. This may be due to optimized sequencing, which may improve the system and its workflow. Additionally, scheduling may improve the process of assigning the available multi-resources of transportation.

Fig.1 shows the flow of WOs within the transportation system. Other units issue each WO to the transportation workshop, and then each WO is checked and scheduled by the transportation unit's planning system. During the process of scheduling, resources (i.e., machine, driver, and rigger) are assigned to the WO and the WO is sequenced. The transportation unit sends both resources and services to the demander unit for the processing order according its demand.

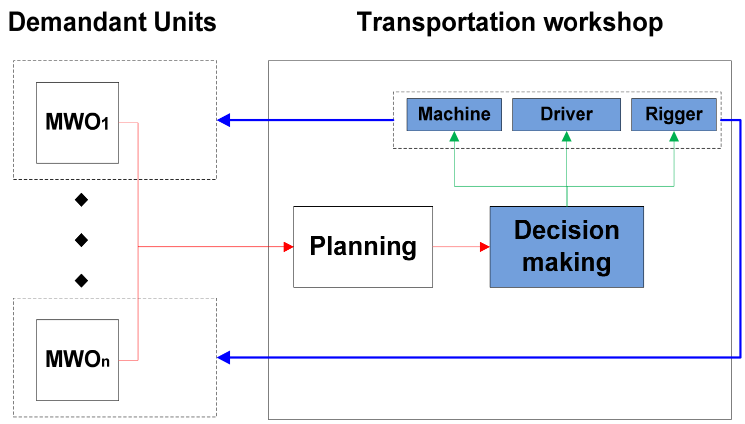

Figure1. Maintenance work orders (MWO) flows in transportation system of the company

In a job shop with m machines, each job is passed through a predetermined route. In this structure, there are not only job shops in which each order is processed on each machine at least once., but also job shops in which an order (including operations) is operated on each machine more than once (Pinedo, 2012). The structure of the transportation workshop is similar to the environment of job shop using the multi-resource assignment. As shown in Fig.2, there is a set of jobs cases JC = {J1,J2…,Jn}, or WO = {WO1,WO2…,WOn} to be operated on the combination of available required resources R = {machine: M1,…, Mm,driver:D1,…, Dd,rigger:R1,…, Rr} in each stage. Since there are some stages for each WO, each job case JCi is formed by a sequence of some operations {Oi,j}. The multi-resource assigned to i-th job for each stage is presented by the blue-boxes over the Ji. The processing time of the Oij on required resources is represented in the work flow. Therefore, in the MRJSP, both allocation of the efficient available resources, and the sequence of job cases on all resources are determined to minimize the makespan (Cmax). For instance, in Fig.2, work ordr #3 in stage #1is notated by WO3-1 and it is processed by machine #2, driver #2, and rigger #1 sequenced by WO1-1.

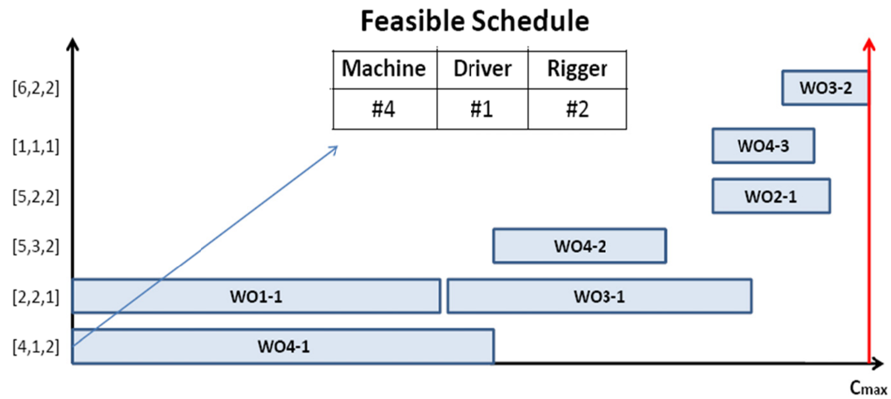

Figure 2. Feasible schedule of work orders in the transportation system of the company

3.2. Mathematical programming for transportation system scheduling

Since the structure of the MRJSP was a combinatorial and an NP-hard problem, the mathematical programming models were unable to provide efficient tools to solve these problems with large sizes. However, they can be considered as the first step to develop an effective heuristic, and evolutionary computation to solve this problem with a large size. The MILP model was constructed by Ozguven et al. (2010) for a flexible job shop scheduling (FJSP). Therefore, we formulated the Ozguven's MILP model for the MRJSP problem because resources in the addressed problem were flexible for the processing orders. Several assumptions were adopted to define daily MRJSP as follows:

1. Only deterministic orders were involved in this study, and release/arrival time was assumed static and equal to zero.
2. All resources (machines, riggers, and drivers,) were always available during the working day and zero time, and there was no resource failure.
3. All resources were assumed to be identical in the processing time.
4. Orders were only allowed to be operated on a subset of machines based on specialty.
5. All drivers were eligible for driving all of the machines.
6. It was assumed that all the riggers accompany all of the drivers in the workshop.

The sets/indices were described in Table 1. In the problem, some of the deterministic jobs in the set *I*, and three resource types in the set *R*, which some types were involved in each stage were defined. To process the work orders, there were three resource types including the machine, the driver, and the rigger. Consequently, some parameters were defined according to Table 2.

Table 1.Indices and sets for MILP model

Sets	Description
$\mathcal{I}$	Set of all the jobs$i, h \in I$
$\boldsymbol{MA}$	Set of machine group
$\mathcal{O}_m$	Subset of orders (job with stage) based on machine group$\boldsymbol{m} \in \boldsymbol{MA}$
$\mathcal{J}_i$	Set of operations of job$i \in I,\ j, g \in J$
$\mathcal{R}$	Set of all resource types
$\mathcal{R}_{ij}$	Subset of capable resource type for operation O_{ij} or stage j job$i \in I, j \in J_i$
$\mathcal{K}_r$	Set of all the resources in resource type r(exception of machine group) $r \in R - \{1\}$, and $K_{r_{ij}}$ is set of resources type$r$ for stage $j$job$i$
$\mathcal{K}_{r_m}$	Subset of all special machine based on group $\boldsymbol{m} \in \boldsymbol{MA}m$

Table 2.Parameters for MILP model

Parameters	Description
P_{ijrk}:	Processing time of operation O_{ij} if performed on resource k of type r
M:	A large positive number
n:	Total number of jobs (work orders)
h_r:	Total number of resources for each resource type (3 types*)

* In this study, 8resource types are introduced that will be explained more in section 4

Applied variables in this mathematical model were divided into decision and auxiliary variables, which were described by the notations in Table 3. As shown, the auxiliary variable $C_i$was employed to calculate makespan as the objective function.

Table 3. Decision and auxiliary variables for MILP model

Variables	Description
Decision variables	
S_{ijrk}:	The start time of operation O_{ij} by resource k of type r
C_{ijrk}:	The end time of operation O_{ij} by resource k of type r
C_{max}:	Makespan
v_{ijrk}:	Equals to 1 if operation O_{ij} performed on resource k of type r, equals 0 otherwise
z_{ijhgrk}:	Equals to 1 if operation O_{ij} precedes operation O_{hg} on resource k of type r, equals 0 otherwise
g_{ijrk}:	Equals to 1 if operation O_{ij} performed by resource k of type r, equals 0 otherwise. This variable is used for other resources exception of the machine (driver, rigger).
Auxiliary variables	
C_i :	The completion time of job i

A general model of MILP was formulated for the MRJSP problem with n jobs as below:

$$\min C_{max} \tag{1}$$

$$s.t.$$

$$C_i \le C_{max} \quad \forall\, i \in \boldsymbol{\mathcal{I}} \tag{2}$$

$$C_i \ge \sum_{k \in K_{r_{ij}}} C_{ijrk} \; \forall\, i \in \boldsymbol{\mathcal{I}}, j \in \boldsymbol{\mathcal{J}_i}, r \in \boldsymbol{\mathcal{R}}_{ij} \tag{3}$$

$$S_{ijrk} + C_{ijrk} \le M v_{ijrk} \quad \forall\, i,j \in \boldsymbol{\mathcal{O}_m}, r \in \boldsymbol{\mathcal{R}}_{ij} = \{1\}, k \in \boldsymbol{\mathcal{K}}_{r_{ij}} \tag{4}$$

$$S_{ijrk} + P_{ijrk} - M(1 - v_{ijrk}) \le C_{ijrk} \quad \forall\, i,j \in \boldsymbol{\mathcal{O}_m}, r \in \boldsymbol{\mathcal{R}}_{ij} = \{1\}, k \in \boldsymbol{\mathcal{K}}_{r_{ij}} \tag{5}$$

$$S_{ijrk} + C_{ijrk} \le M v_{ijrk} \quad \forall\, i \in \boldsymbol{\mathcal{I}}, j \in \boldsymbol{\mathcal{J}_i}, r \in \boldsymbol{\mathcal{R}}_{ij} = \{2,3\}, k \in \boldsymbol{\mathcal{K}}_{r_{ij}} \tag{6}$$

$$S_{ijrk} + \sum_{k \in K_{r_m}} P_{ijrk} g_{ijrk} - M(1 - v_{ijrk}) \le C_{ijrk} \quad \forall\, i \in \boldsymbol{\mathcal{I}}, j \in \boldsymbol{\mathcal{J}_i}, r \in \boldsymbol{\mathcal{R}}_{ij} = \{2,3\}, k \in \boldsymbol{\mathcal{K}}_{r_{ij}} \tag{7}$$

$$C_{hgrk} - Mz_{ijhgrk} \leq S_{ijrk} \quad \forall\, i \ll h, j, g, r \in \boldsymbol{\mathcal{R}}_{ij} \cap \boldsymbol{\mathcal{R}}_{hg}, k \in \boldsymbol{\mathcal{K}}_{r_{ij}} \cap \boldsymbol{\mathcal{K}}_{r_{hg}} \tag{8}$$

$$C_{ijrk} - M(1 - z_{ijhgrk}) \leq S_{hgrk} \quad \forall\, i \ll h, j, g, r \in \boldsymbol{\mathcal{R}}_{ij} \cap \boldsymbol{\mathcal{R}}_{hg}, k \in \boldsymbol{\mathcal{K}}_{r_{ij}} \cap \boldsymbol{\mathcal{K}}_{r_{hg}} \tag{9}$$

$$\sum_{k \in K_{r_{ij}}} S_{ijrk} \geq \sum_{k \in K_{ri(j-1)}} C_{i(j-1)rk} \,\forall\, i \in \boldsymbol{\mathcal{I}}, j \in \{2, \dots, \boldsymbol{\mathcal{J}_i}\}, r \in \boldsymbol{\mathcal{R}}_{ij} \tag{10}$$

$$\sum_{k \in K_{r_{ij}}} S_{ijrk} = \sum_{k\prime \in K_{rij}} S_{ijr'k'} \,\forall\, i \in \boldsymbol{\mathcal{I}}, j \in \boldsymbol{\mathcal{J}_i}, or\, \forall\, (i, j \in \boldsymbol{\mathcal{O}_m}), r, r' \in \boldsymbol{\mathcal{R}}_{ij} \tag{11}$$

$$\sum_{k \in K_{r_{ij}}} C_{ijrk} = \sum_{k\prime \in K_{rij}} C_{ijr'k'} \,\forall\, i \in \boldsymbol{\mathcal{I}}, j \in \boldsymbol{\mathcal{J}_i}, or\, \forall\, (i, j \in \boldsymbol{\mathcal{O}_m}), r, r' \in \boldsymbol{\mathcal{R}}_{ij} \tag{12}$$

$$\sum_{k \in K_{r_{ij}}} v_{ijrk} = 1 \quad \forall\, i \in \boldsymbol{\mathcal{I}}, j \in \boldsymbol{\mathcal{J}_i}, or\, \forall\, (i, j \in \boldsymbol{\mathcal{O}_m}), r \in \boldsymbol{\mathcal{R}}_{ij} \tag{13}$$

$$\sum_{k \in K_{r_m}} g_{ijrk} = 1 \quad \forall\, i, j \in \boldsymbol{\mathcal{O}_m}, r \in \boldsymbol{\mathcal{R}}_{ij} \tag{14}$$

$$S_{ijrk}, C_{ijrk} \geq 0 \qquad \forall\, i, j, r, k \tag{15}$$

$$v_{ijrk}, g_{ijrk} \in \{0,1\} \quad \forall\, i, j, r, k \tag{16}$$

$$z_{ijhgrk} \in \{0,1\} \qquad \forall\, i \ll h, \forall j, g, \forall\, r \in \boldsymbol{\mathcal{R}}_{ij} \cap \boldsymbol{\mathcal{R}}_{hg}, \forall\, k \in \boldsymbol{\mathcal{K}}_{r_{ij}} \cap \boldsymbol{\mathcal{K}}_{r_{hg}} \tag{17}$$

In the above model, Equation (1) states the minimum objective functions,i.e., makespan. Constraint (2) determines the makespan based on the completion time of the work orders. Equation (3) reflects the completion times of the work orders in the last stage. Constraints (4) and (5) guarantee that the difference between the start time and the end time of the order processing on the first resource (i.e machine) is equal to the processing time of that order on the related resource. Constraints (6) and (7) guarantee the same requirements of equations (4) and (5) but for the other involved resources in the second stage exception of machine (i.e. the driver and the rigger). Constraints (8) and (9) were used to ensure that two different operations of O_{ij} and O_{hg} cannot be performed at the same time on any resources in the set $R_{ij} \cap R_{hg}$. Equation (10) specifies that the *j*th operation of each job must be exactly started after the completion time of *(j-1)*th of the operation of the same job. Constraint (11) and (12) determine that all the required resources for each stage must have an identical start time and an identical completion time. Equation (13) requires that one and only one resource from each resource type can be allocated to an operation of the order. Finally, constraint (14) enforces that one and only one machine from the eligible machine group can perform the procedure, and it is possible that the others are idle or allocated to another case. Constraints (15-17) are positive and binary decision variables.

4. Ant colony optimization and its improvement as the solution procedure

Selfish heard is a new operator inspired by a selfish heard theory which was proposed by Hamilton (1971). In this theory, each member of the population (heard) tries to reduce the risk of hunting so that it increases the distance between itself and the hunter. This theory was simulated on the presentation scheme as displayed in Fig.3. Firstly, a point like $x_2$was selected as the predator randomly; then, another point like $x_5$was chosen as the predatin randomly; lastly, a distance of the predation from a predator was changed by applying a coefficient; then, a new point was nx_5 obtained from the updating procedure. In nature-inspired meta-heuristic algorithms, there are two key components including intensification and diversification so that adjusting between these may make efficient optimization technique (Behmanesh, 2016), thus, SH is applied to improve diversification mechanism of ACO for finding better solutions. To implement SH in ACO algorithm, an ant is selected as the predator, and some agents behave as a predatin. Therefore, predatins change their route to increase distance and so this displacement updates the pheromone of the route, randomly. This pheromone updating is effective on diversification.

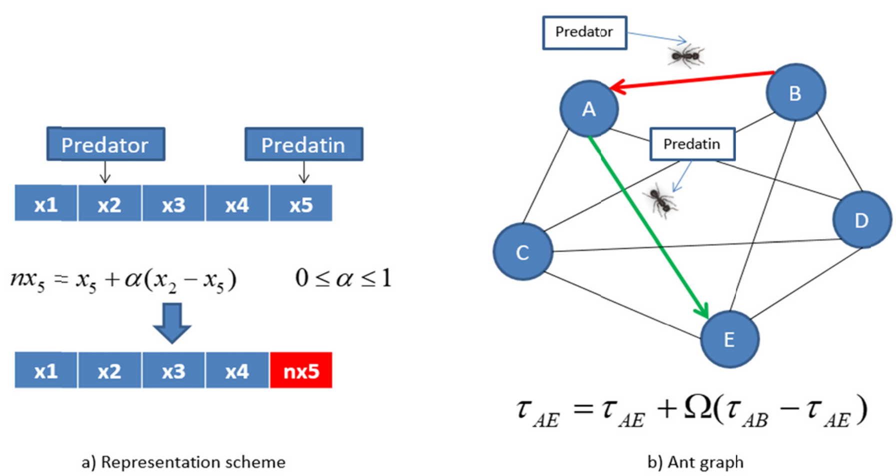

Figure 3. Selfish heard operator on a representation scheme

The two-level ACO algorithm was tailored by mapping cities to work orders, and thereby, the sequence of orders was constructed by a nodes tour. In Xiang's procedure (2015), order cases were sequenced in the outer graph and required multi-resource types were allocated to the orders in an inner graph. The available resources in the same resource type were represented by the nodes in the inner graph. These two graphs are illustrated in Fig.4. The sequence of the orders was determined in Fig.4a., and the required resources (the machine, the driver, and the rigger) were assigned to each order according to Fig.4b.The resources assigned to the work order for each stage was determined based on the path that ant foraged in the inner graph. A mix pheromone update strategy was defined for the algorithm, and it comprised one local and two global minima. In the outer level, the best agent updated the trails based on a global strategy to search the best sequence. In the inner level, the case-related pheromone was defined to save the information that connected work order with the required resource, based on the global strategy, while an inner resource-related was defined to record the information to utilize resource based on a local strategy. It must be noted that the local updating was effective until ant forages path of the inner graph and it was invalid after going out of the inner.

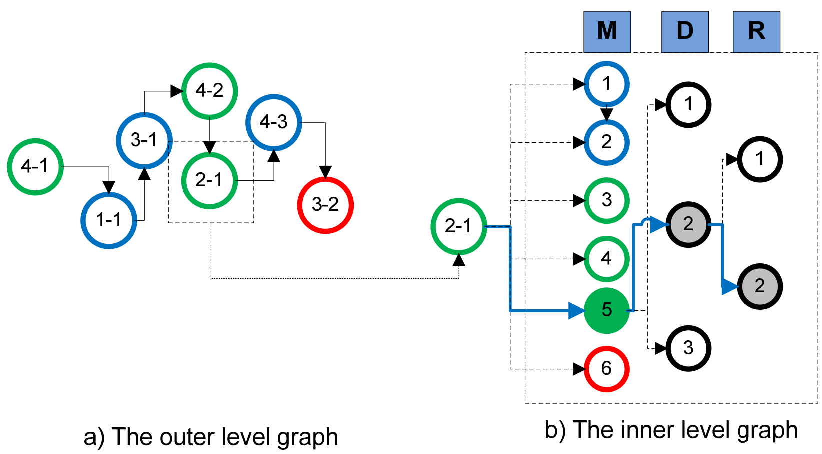

Figure 4. A two-level ant graph for multi-resource job shop (work order scheduling)

Also, the following pseudo code display the updating pheromone strategy based on the selfish heard theory.

Algorithm. Update pheromone (ph) based selfish herd operation (SH)
1. Input: the number of hunted (*nh*) , coefficient of distance(*omega*)
2. Determine the nodes of hunter randomly as (*hi,hj*)
3. **For** *i* = 1: *nh* **do**
4. Determine the nodes of hunted randomly as (*i,j*)
5. ph (*i,j*) = ph (*i,j*) + *omega* * (ph (*hi,hj*) - ph (*i,j*))
5. **End for**
6. Return matrix of pheromone

5. Results and statistical analysis

To evaluate the proposed approaches, we took five test cases. These cases were classified into small, medium, and large which were differed in processing time, the number of the jobs, the maximum order cases for each job, and the allocated resources. Cases category and their specifications were shown in Table 4. Cases were categorized by five types of small, medium, large, and extra-large. Each problem in each case can be generated based on the different structure of the types of the processing time. As it was observed, problems were different in terms of size of the jobs (column 2), the maximum orders for the jobs (column 3), the total orders for each case (column 4), the size of resources (column 5-7), and the case type structure based on the processing time (column 8). In the process of generating data for processing time, we considered ranges in which an interval showed values, and it specified that the given value was selected within this range, randomly.

Table 4. Test cases and structure

problem	Jobs	Max-ord	Orders	Machine	Driver	Rigger	Time
1	10	3	25	6	2	2	[8,100]
2	11	3	27	6	2	2	[13,200]
3	15	3	39	8	5	4	[60,300]
4	20	4	47	8	6	4	[150,300]
5	40	4	93	8	10	8	[180,450]

All algorithms were coded in MATLAB language and ran on an Intel Core (TM) Duo CPU T2450, 2.00 GHz computer with 1 GB of RAM. Moreover, the MIP model was coded in GAMS software and ran by CPLEX solver. Assessing the proposed algorithm was done in this subsection and was divided into two parts; First, the algorithm was validated on the small simulated case in comparison with the MILP model. Then, it was evaluated on the small to large simulated cases in comparison with an ACO without the SH strategy. We first ran the MILP model on the very small case as presented in Table 7 (appendix) along with the small data of Table 4 to validate the proposed approach. The algorithm was repeated 10 times and the mean of makespan found by BWAS+ was compared with the MILP. It must be noted that BWAS+ algorithm was validated in comparison with the MILP model on seven small cases as shown in Table 5. Columns (2-3) show results of the methods, and columns (4-5) display the gap between the result of the two methods, and the computational time of those. The results of the relative objective and time in Table 5 demonstrate that the proposed algorithm is able to obtain 99.45% of the average optimal solution in 12.14% of the average time of the exact algorithm

Table 5. Comparison of the performance of the BWAS+ and the MILP

Sample no.	MILP	BWAS+	GAP (%)	CT(BWAS+/ MILP)
1	**100**	**100**	0.00%	3/5
2	**125**	**125**	0.00%	4/35
3	**172**	**172**	0.00%	5/1000
4	**216**	**216**	0.00%	7.3/2000
5	**256**	**256**	0.00%	8.8./3000
6	**293**	**303**	3.3%	10/3600
Average			**0.55%**	**0.1214**

After determination of the best setting parameter for each algorithm, all algorithms were repeated 20 times for each instance tested for comparison. Since the makespan values of the problems were heterogeneous, we applied the ratio percentage deviation (RPD) index to homogenize all the data. The RPD value of each makespan was obtained according to the following equation:

$$RPD_{ij} = \frac{makespan_{ij} - min_j(makespan_{ij})}{min_j(makespan_{ij})} \tag{18}$$

where the index of the problem was notated by i and j.

To compare all the algorithms with or without the SH strategy for solving problems, we ran them on a dataset in Table 4. The results of the normalized experiments based on the RPD were indicated in Table 6 and also ANOVA test using the least significant differences intervals (99% confidence level) of the makespan for comparison of all the algorithms were done and the results indicated that the makespan obtained by the algorithms were significantly different. Therefore, the solutions obtained by the algorithms with the SH outperform all of the algorithms without the SH strategy. Consequently, we inferred that the new proposed approach with considering the SH is a promising meta-heuristic algorithm to provide good solutions for solving MRJSP problems. A significant difference between the mean of the makespan for the algorithms with or without the SH strategy on five sample cases was shown in Fig.6. Besides, the boxplot of makespan for all algorithms with / without SH strategy on fifth sample (large case) is displayed in Fig.5.

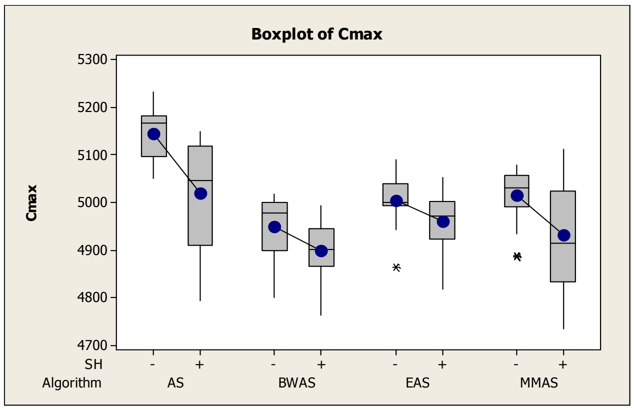

Figure 5.The boxplot of Cmax for AS, MMAS, BWAS, and EAS algorithms with/without SH

Horizontal axis shows using SH strategy, and algorithm types in two rows respectively, and vertical axis shows objective value. It is inferred from boxplot that each algorithm with SH (+) outperforms same algorithm without SH (-). And results indicate that applying SH strategy in proposed algorithms improves diversification mechanism and perform better solutions. Fig.6 presents the results of the normalized experiments and indicates that the BWAS+ outperforms the other ant colony optimization algorithms.

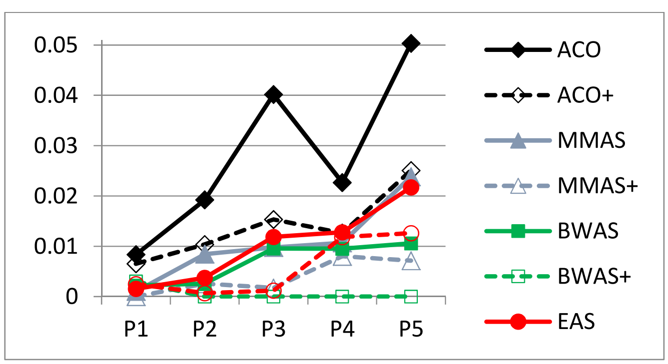

Figure 6.The quality of solutions obtained by AS, MMAS, BWAS, and EAS algorithms with/without SH for all cases

Table 6. Results of normalized experiments

Case	AS	AS+	MMAS	MMAS+	BWAS	BWAS+	EAS	EAS+
1	0.008	0.006	0.001	0	0.002	0.003	0.001	0.002
2	0.019	0.010	0.008	0.002	0.002	0	0.004	0
3	0.040	0.015	0.009	0.002	0.009	0	0.012	0.001
4	0.023	0.013	0.010	0.008	0.009	0	0.013	0.011
5	0.050	0.025	0.024	0.007	0.010	0	0.022	0.013

6. Conclusion

In this paper, we addressed a solution for multi-resource job-shop scheduling problem (MRJSP) and proposed a novel approach to optimize this problem with a combinatorial nature. A real environment in the transportation system of a company was taken into account as an MRJSP case. Some units required services from the transportation to complete their required maintenance. Therefore, three resource types including the drivers, the riggers, and the vehicles were assigned to a work order simultaneously. The unique contribution of this paper was to introduce a new pheromone updating strategy, SH, to improve the quality of the ACO algorithms to solve the MRJSP problem. Then, four ACO algorithms including AS, MMAS, BWAS, and EAS were applied to find the near-optimal schedule. To illustrate the new ACO with the SH strategy, five test data were generated in which all cases were different in the processing time, the numbers of the jobs, the maximum orders of cases for each job, and the allocated resources. Following results and discussions, it can be concluded that the improvement over the four basic ACO algorithms was effective significantly and using the SH strategy in algorithms made a new method that can outperform traditional ACO algorithms for solving the MRJSP problems. The results indicated that the proposed strategy enhanced the diversification mechanism of the algorithms to find more quality solutions.

Ultimately, we suggest some directions to extend the ACO algorithm as opportunities for the future work in this area. To construct a robust work order schedule, it would be essential to consider an uncertain processing time. In addition, an online case was taken into account as the problem. Consequently, a new ACO algorithm was constructed to solve the online order scheduling in the real world. On the other hand, building a new ACO algorithm for the multi-objective MRJSP can be a novel work in the future research. Therefore, the future research can be an extension to multi-objective ACO algorithm for solving the multi-objective MRJSP problems using the fuzzy processing time.

REFERENCES

[1] **Asadzadeh, L., Zamanifar, K. (2010),** ***An Agent-Based Parallel Approach for the Job Shop Scheduling Problem with Genetic Algorithms;*** *Mathematical and Computer Modelling*, 52, pp. 1957_1965;

[2] **Artigues, L. Lopez, P. Ayache, P. D. (2005),** ***Schedule Generation Schemes for the Job-Shop Problemwith Sequence-Dependent Setup Times: Dominance Properties and Computational Analysis;*** *Annals of Operations Research*,138, pp. 21–52;

[3] **Augusto, V., Xie, X., Perdomo, V. (2010),** ***Operating Theatre Scheduling with Patient Recovery in both Operating Rooms and Recovery Beds;*** *Computers & Industrial Engineering*, Vol. 58(2), pp. 231–238;

[4] **Beck, J. C., T. Feng and J.-P. Watson (2011),** ***Combining Constraint Programming and Local Search for Job-Shop Scheduling;*** *INFORMS Journal on Computing, 23(1): 1-14;*

[5] **Behmanesh, R. (2016),** ***Nephron Algorithm Optimization : Inspired of the Biologic Nephron Performance;*** *International Journal of Applied Metaheuristic computing,* Vol. 7(1), pp. 38-64;

[6] **Behmanesh, R., Zandieh, M., & Hadji Molana, S. M. (2020),** ***Multiple Resource Surgical Case Scheduling Problem: Ant Colony System Approach;*** *Economic Computation & Economic Cybernetics Studies & Research*, *54*(1).;

[7] **Dorigo, M., Maniezzo, V., Colorni, A. (1991),** ***Positive Feedback as a Search Strategy*;** *Technical Report 91-016, Dip. Elettronica*, Politecnico di Milano, Italy;

[8] **Ferland JA, Ichoua S, Lavoie A, Gagne E. (2001),** ***Scheduling Using Tabu Search Methods with Intensification and Diversification;*** *Computers & Operations Research*, 28, pp.1075–1092;

[9] **Grimes, D. and E. Hebrard (2015),** ***Solving Variants of the Job Shop Scheduling Problem through Conflict-Directed Search;*** *INFORMS Journal on Computing,* 27(2): 268-284;

[10] **Guo, Z.X. Wong, W.K, Leung,S.Y.S. Fan, J.T. Chan S.F. (2006),** ***Mathematical Model and Genetic Optimization for the Job Shop Scheduling Problem in a Mixed- Andmulti-Product Assembly Environment: A Case Study Based on the Apparel Industry;*** *Computers & Industrial Engineering*, 50, pp. 202–219;

[11] **Hamilton, W.D. (1971),** ***Geometery for Selfish Heard;*** *J.theor.Biol*, 31, pp. 295-311;

[12] **Heinonen, J., Pettersson, F., (2007),** ***Hybrid Ant Colony Optimization and Visibility Studies Applied to a Job-Shop Scheduling Problem*;** *Applied Mathematics and Computation*, 187(2), pp. 989-998;

[13] **Huang R.H., Yang, C.L. (2008),** ***Ant Colony System for Job Shop Schedulingwith Time Windows*;** *Int J Adv Manuf Technol*, 39, pp. 151–157;

[14] **Kundakci, N. Kulak, O. (2016),** ***Hybrid Genetic Algorithms for Minimizing Makespan in Dynamic Job Shop Scheduling Problem;*** *Computers & Industrial Engineering*, 96, pp. 1–51;

[15] **Liu, S.Q. Kozan, E. (2016),** ***Parallel-identical-machine Job-shop Scheduling with Different Stage-Dependent Buffering Requirements;*** *Computers & Operations Research*, 74,pp.31–41;

[16] **Mati, Y., Xiaolan Xie, X. (2008),** ***A Genetic-Search-Guided Greedy Algorithm for Multi-Resource Shop Scheduling with Resource Flexibility*****;** *IIETransactions*, Vol. 40(12), pp. 1228-1240;

[17] **Mati, Y. (2010),** ***Minimizing the Makespan in the Non-Preemptive Job-Shop Schedulingwith Limited Machine Availability;*** *Computers & Industrial Engineering*, 59, pp. 537–543;

[18] **Pan, J. C. H., Huang. H. ch. (2009),** ***A Hybrid Genetic Algorithm for No-Wait Job Shop Scheduling Problems*****;** *Expert Systems with Applications*, 36, pp.5800–5806;

[19] **Pinedo, M. (2012),** ***Scheduling: Theory, Algorithms, and Systems*****;** *Springer-Verlag*, New York, 4th edition.

[20] **Salido, M.A.Joan, E. Barber, F. Giret, A. (2017),** ***Rescheduling in Job-shop Problems for Sustainable Manufacturing Systems;*** *Journal of Cleaner Production*, 162, pp.121-132;

[21] **Tiwari, P.K., Vidyarthi, D.P., (2016),** ***Improved Auto Control Ant Colony Optimization Using Lazy Ant Approach for Grid Scheduling Problem*****;** *Future Generation Computer Systems,*Vol. 60, pp. 78-89*;*

[22] **Xiang, W., Yin, J., Lim, G. (2015),** ***An Ant Colony Optimization Approach for Solving an Operating Room Surgery Scheduling Problem*****;** *Computers & Industrial Engineering*, Vol. 85, pp. 335–345.

Appendix

Table 7. Test cases and structure for MILP model

problem	Jobs	Orders	Machine	Driver	Rigger	Time
1	3	8	6	2	2	[8,100]
2	4	9	6	2	2	[8,100]
3	5	12	6	2	2	[8,100]
4	6	15	6	2	2	[8,100]
5	7	17	6	2	2	[8,100]
6	8	20	6	2	2	[8,100]