

# Integrated hybrid flow shop scheduling and vehicle routing problem

# Raheleh Moazami Goodarzi<sup>1</sup>, Fardin Ahmadizar<sup>1\*</sup>, Hiwa Farughi<sup>1</sup>

<sup>1</sup> Industrial engineering department, University of Kurdistan, Sanandaj, Iran

moazami@eng.uok.ac.ir, f.ahmadizar@uok.ac.ir, h.farughi@uok.ac.ir

#### **Abstract**

In this paper, a new integrated mathematical model for production and distribution planning is presented to minimize tardiness and transportation costs. A mixedinteger linear programming (MILP) formulation is developed for the problem which consists of two parts. First, the production scheduling in a hybrid flow shop (HFS) environment with identical machines in each stage, and then, the delivery of completed jobs with a fleet of vehicles that have the same capacity. Due to the NPhard nature of the problem, a new metaheuristic approach based on Particle Swarm Optimization Algorithm (PSO) and Genetic Algorithm (GA) is presented to solve the integrated problem. GA's operators are used to update the particle position of the PSO algorithm. The algorithm uses dispatching rules to represent the initial solution and searches in the solution space including active schedules. To investigate the efficiency and effectiveness of the proposed method, numerical studies are carried out with random problems. The computational results show that the proposed solution approach yields fairly good results in comparison with the PSO versions in the subject literature. The algorithm is capable of generating relatively good solutions for sample cases.

**Keywords:** Integrated production and distribution scheduling, hybrid flow shop, vehicle routing problem, particle swarm optimization algorithm, genetic algorithm.

#### 1-Introduction

There is intense competition for companies that are influenced by their customer demands to find heuristic methods as keeping costs down and staying competitive, while they are providing excellent customer service. Due to this global environment complexity, scheduling and transportation problems not only can reduce costs but also ensure the service quality and satisfaction in the business. The HFS environment is one of the development problems of the scheduling and sequencing models, in which various production stages are grouped in series, and jobs are processed on the machines respectively. Each component of each product can be manufactured by a machine, and if there are more than one particular machine which are located in a stage as parallel lines, the subject of machine allocation will also arise. Tardiness costs play a pivotal role in production costs in a HFS production environment; in fact, transportation costs have a significant share in the final cost of goods. Therefore, the optimization of the distribution system is one of the main goals of the companies. In the real world, where orders size is smaller than the vehicle's capacity, allocating one or more orders to a vehicle is possible. Vehicle routing will be used to minimize tardiness cost and transportation costs.

\*Corresponding author

ISSN: 1735-8272, Copyright c 2021 JISE. All rights reserved

In classic supply chain management, various activities optimization, such as the purchase of raw materials, and the production and distribution of the finished products, are carried out individually. However, with the improvement of optimization methods and increasing the computing speed, it is now possible to use integrated models and simultaneously plan the levels of the supply chain. One of the most important advantages of integrated models is the possibility of using the constraints and the various objective functions, which lead to a variety of proposed models. In the real world and real goods production processes such as foods, pharmaceutical, and medical production, chemical and petrochemical production, ceramic, clothes, semiconductors, electronics manufacturing, and airplane engine production, not only their production environment being an HFS, due to the aforementioned reasons, consideration of the vehicle routing while solving the production scheduling problem can reduce production and distribution costs specially tardiness costs. Therefore, it seems necessary to consider integrated HFS and VRP models together.

In this paper, a mixed-integer programming (MILP) model is presented that includes production scheduling, and the goal is to optimize an integrated scheduling problem for the HFS and the distribution and vehicle routing at the operational level. In the literature, scheduling and routing problems often deal with problems with the single-machine production environment or parallel machines and examine the flow shop (FS) environment with single-machine processes. According to our knowledge, this is the first attempt to consider the HFS scheduling with parallel identical machines at each stage, and VRP together. The vehicle capacity constraints are also considered for the first time in an integrated scheduling environment for HFS and VRP. The goal is to minimize the tardiness costs of delivering the client's job and transportation costs. To solve the integrated problem, a hybrid PSO algorithm is used. The proposed algorithm is a combination of the PSO algorithm and the GA algorithm, and the GA's operators are used to update the particle position of the PSO (Pan et al, 2008). The algorithm also used the EDD and ERT rules for the display of the initial solution section and searches in the solution space including active schedules. Finally, numerical analysis is used to show the performance of the proposed algorithm.

The paper is organized as follows: In the next section, the literature on integrated production scheduling and distribution models will be reviewed. In section 3, the problem considering is fully defined. In section 4, a metaheuristic algorithm is proposed to solve the mathematical model and in section 5, the computational results are presented. Eventually, in the final section, the ultimate conclusions and topics for future research will be presented.

#### 2-Literature review

An overview of the most important models of mathematical planning for HFS scheduling and vehicle routing is presented. Based on the review of existing studies, a framework for the production scheduling and vehicle routing consists of three main steps: the problem definition and the production system characteristics, the model design, and the solution method.

Karimi and Davoudpour (2015) have scheduled the production and distribution of the supply chain with related factories, including the supplier and the manufacturer. These components of the chain are grouped in series. The jobs move in batches. The number of categories is not limited, but the capacity of each category is limited. The goal is to balance the transportation cost and the tardiness penalty. The branch and bound method is presented for the problem. Chieu Ta et al. (2015) have presented the problem of integrated FS scheduling with multiple machines and distribution and vehicle routing by studying a case in the real world and the chemotherapy industry. Only an unlimited capacity vehicle is available for delivery. The goal of the problem is to minimize total tardiness. A greedy heuristic algorithm based on a taboo search is proposed to solve the problem. Rohmer and Billaut (2015) have considered the integrated production scheduling problem with the two factors of the manufacturer and the logistics company. This production environment is a FS and the purpose of the problem is to minimize a cost function consisting of inventory costs, vehicle costs, and tardiness penalties. Vehicles have a fixed departure date. Their number and capacity are unlimited. Heuristic algorithms are based on the due date and the search for neighborhood solutions). Johar et al. (2016) have proposed a production and distribution planning problem. The objective function involved minimizing the total weight of transportation costs and the

tardiness penalty. The problem posed by Lacomme et al. (2016) involved the integrated production scheduling problem and transportation with limited capacity and short lifespan products with several vehicles. Armstrong et al. (2008) presented a problem of producing and distributing perishable materials with single machine production and a vehicle with time-window; the goal is to find a subset of customers that can help maximize demand-supply. Alvarez et al. (2015) suggested a multifactor approach for production planning with parallel machines and dynamic distribution with time-windows in a supply chain.

Amorim et al. (2013) determined the size of production batches in job shop production for systems in which they are considered to be perishable products. In this research, production is carried out in the environment of parallel vehicles and carriers along with the routing of similar vehicles with limited capacity. Chen et al. (2009) presented a tactical and operational nonlinear mathematical model taking into account the production planning and vehicle routing for perishable food products in a two-stage, single-period, multi-retailer form. Low et al. (2014) have presented an integer nonlinear programming model and two adaptive genetic algorithms for the problem of single-machine production and routing of heterogeneous vehicles. Ulrich (2013) merged production and distribution schedules to minimize total tardiness. Kumar et al. (2015) considered a VRP that simultaneously examined the problem of single machine production and routing with limited-capacity vehicles. Ramezanian et al. (2017) have studied the simultaneously planning of production and delivery operations as major and important operations in the production system with an integrated view. Moons et al. (2017) focused on production and distribution planning problems that explicitly address the decision of vehicle routing in the delivery process. The literature on integrated production planning and VRP are reviewed and categorized.

As mentioned, the goal of this paper is to integrate the HFS scheduling, distribution, and VRP following the objectives of minimizing production and distribution costs. In the literature, scheduling and routing problems often deal with problems with a single-machine production environment or with parallel machines. In this paper, is attempting to consider the HFS scheduling with parallel machines at each stage, and integrating its scheduling with limited capacity vehicle routing. This problem has not been studied so far, and the need to consider it is felt. In this paper, a MILP model is presented for production scheduling in the HFS environment. At each production stage, several identical machines work in parallel with each other. The jobs enter the first stage with prioritization, and they pass the production stages accordingly. Machines are not idle until a job is available. After the production completion, the jobs are delivered to the customers using limited capacity vehicles. The purpose of the model is to minimize tardiness costs and transportation costs.

Given that the problem of the HFS is included in NP-hard problems (Gupta, 1988), as well as the vehicle routing problem is included in NP-hard problems (Prince, 2004), so the integrated problem, which includes production scheduling in the HFS environment and vehicle routing, also is NP-hard, and if the problem dimensions are large, it is not possible to reach the global optimum in a reasonable time. To find the solution for the integrated problem in less time, a combination of PSO and GA algorithms is proposed, and the GA's operators are used to update the particle position of the PSO algorithm. The advantages of the PSO algorithm include a simple structure, immediately accessible for practical applications, easy implementation, speed to acquire solutions, and robustness that are sustained in the literature. Also, GA is one of the most popular metaheuristic algorithms which have been used in many combinatorial optimization problems. The main property of this algorithm is related to the knowledge sharing between individuals in the population using a crossover operator. As a general solution method, GA is extensively applied to solve most of the scheduling and routing problems (Diveev and Bobr, 2017). The proposed algorithm uses the EDD and ERT rules in the representation part of the initial solution and searches in the solution space including active schedules. This algorithm is presented for the first time in the literature of scheduling problems. Finally, numerical analysis is performed to evaluate the performance of the proposed algorithm, compared to three versions of PSO, IPSO, and the results are examined.

# 3-problem definition

The assumptions, notations, parameters, variables and model of the problem are defined as follows. Given the notation is provided by Chen (2010), we define the problem as HFm  $\parallel$  V ( $\infty$ , K), routing  $\mid n \mid \gamma$ , in which the field  $\alpha$  represents the general HFS scheduling problem and there are m machines at each production stages. The field  $\beta$  is empty because no constraint is defined in the schedule. The next section includes V ( $\infty$ ,K) and routing, which indicates the vehicles routing in the distribution sector. Vehicles have capacity constraints. The number of vehicles that are used for delivery is unlimited and they are available to any extent required. Each vehicle has the same fixed capacity and cost and has the same variable cost (homogeneous), while in articles related to the subject of FS and routing, the number of vehicles or capacity is considered unlimited. The next section, fill with the n sign, reflects the fact that each job belongs to a customer. For each job, delivery time is set, after which the tardiness penalty, which is a time-dependent linear function, is imposed on the production and distribution system. All jobs and machines are available at zero time and one job cannot be processed simultaneously on more than one machine. The goal is to minimize total transportation costs and tardiness penalties.

# 3-1-Indices, parameters, decision variables Indices

```
i: Index of the demand points and the number of nodes in the network (i=1,2,...,N) k: Index of serial production stages (k=1,2,...,K) m: Index of machines available at each stage (m=1,2,...,M) j: Index of jobs (j=1,2,...,J) v: Index of vehicles available (v=1,2,...,V)
```

#### **Parameters**

 $p_{i,k}$ : The processing time of the *j*th job on stage k.

 $t_{i,i'}$ : Travel time between node i and node i'(this value is proportional to the distance between the two nodes i and node i').

 $c_{i\,i'}$ : The travel cost between node *i* to node *i'*.

 $d_i$ : Allowed delivery time of the *j*th job to the customer.

 $pi_i$ : Tardiness penalty in delivering job i to the customer.

fc: The fixed cost of using vehicle v.

 $cap_v$ : The capacity of vehicle v (the maximal number of customers that can be serviced during the trip).

 $M_1$ : A number is large enough in manufacturing constraints.

 $M_2$ : A large number.

#### **Variables**

Z: The objective function of the problem including minimizing the tardiness penalties and fixed and variable of transportation costs.

 $co_{i,k}$ : The completion time of the *j*th job in stage *k*.

 $ar_{i,v}$ : The arriving time vehicle v to ith customer node.

 $Rels_v$ : The release time of vehicle v (The completion time of all vehicles' orders)

 $f_j$ : The completion time of the *j*th production in the production line.

 $L_i$ : Tardiness in the delivery of *j*th job to the customer.

 $uu_i$ : The auxiliary variable of the constraint of elimination sub-tour.

 $x_{j,j',k}$ : A binary variable that is equal to 1 when job j at the kth production stage is produced before the job j', otherwise, it is zero.

 $y_{j,k,m}$ : A binary variable that is equal to 1 when job j at the kth production stage is produced on machine m, otherwise it is zero.

 $z_{i,i',v}$ : The binary variable that equals 1 when the vehicle v of the ith node goes to the node i', otherwise, it is zero.

 $u_{\nu}$ : The binary variable that is equal to 1 when using the vehicle  $\nu$ , and otherwise equals to zero.  $w_{i,\nu}$ : The binary variable that is equal to 1 when node i is on the route of vehicle  $\nu$ , otherwise it is zero.

#### 3-2-Mathematical modeling

As mentioned, the purpose of this paper is to plan to schedule the production, distribution, and vehicle routing to minimize production and distribution costs. The proposed problem is the integrated scheduling of the HFS and VRP. In the following, the HFS model with parallel machines at each stage and integrating its scheduling with vehicle routing of limited capacity is presented.

$$Z = \sum_{v} (fc. u_v) + \sum_{i,i',v} (c_{i,i'}. z_{i,i',v}) + \sum_{i} (pi_i. L_i)$$
(1)

$$f_j \ge co_{j,k}$$
  $\forall j, k$  (2)

$$\sum_{m} y_{j,k,m} = 1$$

$$\forall j, k$$

$$(3)$$

$$co_{j,k} - co_{j,k-1} \ge \sum_{m} (y_{j,k,m} \cdot p_{j,k})$$

$$\forall j, k$$

$$(4)$$

$$M_{1}.\left(2 - y_{j,k,m} - y_{j',k,m} + x_{j,j',k}\right) + co_{j,k} - co_{j',k} \ge p_{j,k}$$
  $\forall j, j', j < j', k, m$  (5)

$$M_{1}.(3 - y_{j,k,m} - y_{j',k,m} - x_{j,j',k}) + co_{j',k} - co_{j,k} \ge p_{j',k}$$
  $\forall j, j', j < j', k, m$  (6)

$$\sum_{i'(i'>1,i'\neq i)} Z_{i,i',v} \le u_v$$

$$\forall i, 1 < i < N, v$$

$$(7)$$

$$\sum_{i'(i'\neq i,i'< N)} z_{i,i',v} = u_v$$

$$\forall i, i = 1, v$$
(8)

$$\sum_{i(1 < i < N)} z_{i,N,v} = u_v$$

$$\forall i', i' = N, v$$
(9)

$$\sum_{i'(i'\neq i)} \sum_{v} z_{i,i',v} = 1$$
  $\forall i, 1 < i <$  (10)

$$\sum_{i(i < N, i' \neq i)} z_{i,i',v} - \sum_{i(i > 1, i' \neq i)} z_{i',i,v} = 0$$

$$i' < N, v$$
(11)

$$z_{i,i',v} + z_{i',i,v} \le 1$$

$$\forall i, i', v$$

$$(12)$$

$$z_{i,i',v} = 0$$
  
 $\forall i, i', v, i = N, i' = 1$  (13)

$$uu_i - uu_{i'} + (N.z_{i,i',v}) \le N - 1$$
 (14)  
 $\forall i, i', v$ 

$$\sum_{i'} z_{i,i',v} \le w_{i,v}$$

$$\forall i, v$$
(15)

$$\sum_{v} w_{i,v} = 1$$

$$i < N$$
(16)

$$\sum_{i \neq 0} w_{i,v} \le cap_v \cdot u_v \tag{17}$$

$$ar_{i,v} \ge Rels_v$$
 (18)  $\forall v, i = 1$ 

$$Rels_{v} \ge f_{i} - M_{2}.(1 - w_{i,v}) \tag{19}$$

$$ar_{i,v} \le M_2. w_{i,v} \tag{20}$$

$$L_j \ge ar_{j,v} - d_j$$

$$\forall i, j, i = j + 1, v$$

$$(21)$$

$$\sum_{v} ar_{i',v} \ge (ar_{i,v} + t_{i,i'}) \cdot z_{i,i',v}$$

$$\forall i, i', i' = 1$$
(22)

$$Z, co_{j,k}, ar_{i,v}, f_j, L_j, uu_i \ge 0$$

$$\forall i, i', v, j, k$$

$$(23)$$

$$x_{j,j',k}, y_{j,k,m}, z_{i,i',v}, u_v, w_{i,v} = 0,1$$

$$\forall i, i', v, j, j', k, m$$
(24)

The objective function is described in the constraint (1) which includes the minimization of the tardiness penalties and fixed and variable transportation cost. Constraint (2) ensures that the completion time of job production is higher than the processing time of the job at each stage. The constraint (3) ensures that the job on each stage is exactly processed on a machine of parallel machines in that stage. Constraint (4) ensures that the completion time of the job j in stage k is greater than the completion time of this job in the previous stage plus the processing time of the job in stage k. Constraints (5) and (6) are to establish an appropriate sequence of jobs in the workshop production stage. If job j and j' are proceeding on the same machine at the same stage  $(y_{j,k,m} = y_{j',k,m} = 1 \text{ and } x_{j,j',k} = 1)$ , so the job j has proceeded before j'. According to (7) and (8), the two vehicles do not meet a customer node (except for origin and destination) and each vehicle is considered as a used vehicle if it visits at least one customer node. According to the constraint (9), each vehicle does not meet the destination node more than once. According to the constraints (10), each customer node is visited by exactly one of the vehicles. According to the constraint (11), the number of arcs entering each node is equal to the number of arcs leaving that node (except for origin and destination node). According to the constraint (12), a vehicle does not pass an arc more than one time  $(z_{i,i',v})$  and  $z_{i',i,v}$  can not value 1 simultaneously). According to the constraint (13), there is no arc from the destination to the depo. Constraint (14) is the sub-tour elimination constraint for the vehicle routing problem. Constraint (15) ensures that, as long as the node is not assigned to a vehicle, no edges including that node are passed that vehicle. According to the constraint (16), each customer node is assigned to only one vehicle. According to (17), the orders assigned to each vehicle are as high as that vehicle's capacity. According to the constraints (18) and (19), the arrival time of each order to the destination by any vehicle is larger than the completion time of the production of all orders of that vehicle. That is, every vehicle is not allowed to move until the production of all assigned orders is completed. According to the constraint (20), the vehicle's arrival time is calculated only for the nodes assigned to it. Constraint (21) is used to calculate the tardiness in delivering the job to the customer, which is obtained by differentiating the allowed due date from the delivery time of that job. According to the constraint (22), the time to reach a node is equal to the time it takes for the last edge to reach that node, plus the time it takes to reach the previous node in the vehicle's route. Finally, constraint (23) is related to the positive variables of the problem and the constraint (24) to the binary variables of the problem.

# 4-Proposed hybrid PSO algorithm

As stated in the previous sections, the integrated problem that includes production scheduling in the FS environment and distribution routing is NP-Hard, and if the problem dimensions are large, it is not possible to reach the global optimal at acceptable times. Therefore, for these types of problems, it is necessary to develop methods that will respond faster and sooner. For this purpose, a metaheuristic PSO is used to find the integrated problem solution.

## 4-1-Introduction of PSO algorithm

The PSO algorithm is introduced by Kennedy and Eberhart (1995). This algorithm combines a local search considering the individual experience and a global search considering collective experience, thus demonstrating the efficiency of the search.

Although the PSO algorithm has been proven to be effective in solving hybrid optimization problems, there are many problems in which simple particle optimization algorithms are not able to find optimal or near-optimal solutions at a reasonable and logical time. A disadvantage of the PSO algorithm is that in a large space, it easily reaches the local optimum and has a high convergence rate in the process of iteration. Therefore, a variety of hybrid methods has been proposed to improve the performance of the PSO algorithm. To avoid early convergence, most research on this algorithm focuses on diversity in the search for better configurations that allow the algorithm to escape from the local minimum. Improvement can be divided into two categories: the velocity equation and neighborhood topology. For the velocity equation, Shi and Eberhart (1998) introduced an inertia coefficient for the balancing of general and local agents. Eberhart and Shi (2000) also showed that the use of a decreasing factor is better than the inertial coefficient. Kennedy et al. (2001) used a constant to limit the speed increase to avoid over-speeding. Also, He et al. (2004) introduced a social behavior of influential societies for the velocity equation. The discrete PSO algorithm is first proposed by Kennedy and Eberhart (1997).

For scheduling problems, Tasgetiren et al. (2004) have proposed a continuous PSO to solve single-machine and FS problems. On the other hand, Liao et al. (2007) developed a PSO algorithm based on discrete PSO for the FS scheduling problem. Pan et al. (2008) proposed a discrete PSO algorithm for the FS scheduling problem in which the initial population is created using the NEH neighborhood approach and used the local search based crossover and mutation operators. In their paper, they presented a new method for the particle velocity equation. Santosa et al. (2017) have provided a discrete PSO algorithm for the multi-objective HFS scheduling problem with a waiting time limitation. Jamrus and Chien (2018) have presented a hybrid discrete PSO algorithm with genetic algorithm operators for the job shop scheduling problem with uncertain processing time. Choudhary and et al. (2019) have presented a PSO with the mutation operator to solve the flexible job-shop scheduling problem (FJSP). The target of this paper is to reduce the makespan.

#### 4-2-EDD-ERT-Active-PSO-GA algorithm (EAPG)

To prevent premature convergence and to escape from local optimal, in this paper, genetic algorithm operators that increase diversity the search process is used to refine the particle position in PSO. This hybrid algorithm is a discrete algorithm, which can be used to solve the problem considering the discrete solution space for sequencing, scheduling, and routing problems.

In this paper, a local optimization method is added to this hybrid PSO algorithm to create a hybrid PSO algorithm for the HFS planning problem. In the proposed approach (we called it EAPG), certain operators are introduced to improve the accuracy and efficiency of the algorithm and suggested methods for local searches. As you know, the HFS scheduling problem has a very large solution space, and reducing the

size of the solution space without removing the optimal solution can be very beneficial. Therefore, the corresponding algorithm utilizes the EDD and ERT algorithms in the representation of the initial solution and searches in the solution space including the actual schedules. Finally, numerical analysis is performed to evaluate the performance of the proposed algorithm and the results will be reviewed. The structure of the proposed algorithm is as follows:

- **Step 1** (Generate Initial population): Creates a set of solutions for the first population.
- Step 2 (Assessment): Calculate the objective function for each solution into the population.
- **Step 3** (crossover): crossover operator with a random probability between members of the population and the best individual and global solution.
- Step 4 (Mutation): Apply mutation operators to random particles on the population.
- **Step 5** (Local exploration search): Change the percentage of solution selected from the population using a local search exploration method.
- **Step 6** (Update): Update the best solution and global solutions.
- Step 7 (End): Repeat steps 2- 6 until the ending criteria are fulfilled.

#### **4-3-Solution representation**

In this paper, we use the job-based representation and sequencing method in the first stage (Oguz and Ercan, 2005) (referred to as JBRF in this article). In the job-based approach, the sequence of jobs is displayed in one stage with a simple permutation of jobs. According to this sequence, using vehicle assignment rules, jobs are assigned to different machines at one stage, and in fact, each job is allocated to the first free machine.

Also, the vehicle routing problem requires the identification of two indicators and the adoption of two decisions: the number of vehicles that are needed and the assignment of each job to a vehicle in each stage to the customer. Alba and Dorronsoro (2004), in the solution representation for the routing problem, developed a permutation of random integers in the interval[1, J + V - 1], which includes both customers and path separators, thus allocating customers to the paths. J is the number of customers and V is the number of vehicles. In this permutation, customers are shown with numbers 1 to J and path separators are represented in the interval [J + I, J + V - I].

In this paper, a two-part vector is used to represent the solution, in which the first part includes the job sequence in the first stage of production, and the second part includes vehicle routing.

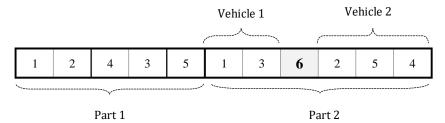


Fig1. Two-part vector for the solution representation of the problem

In the solution, representation is shown in Fig. 1, the permutation 1-2-4-3-5 is set for the jobs. In this paper, the representation method of the solution based on the job is used for the first stage, which is shown in the first part of the solution representation vector, and the jobs are thus assigned to the first free machine until the production stages are completed and entered the transport stage to enter the customer's location. Also, according to the second part of the solution vector, in the sequence above, the first and third work will be carried by the first vehicle, and the second, fifth and fourth jobs will be carried by the second vehicle. Because for dividing *J* between *V* vehicles, *V-1* separator is needed, dividing five customers between two vehicles requires one separator. The numbers that are larger than the number of customers are separators. In the above example, we have five customers, so number 6 is a separator. The numbers between the separators are the customers assigned to the vehicles.

#### 4-4-Decoding process

Since the HFS scheduling problem has a very large search space, reducing the size of the solution space without neglecting the optimal solution is very beneficial. The proposed heuristic algorithm in this paper is in the active scheduling space, which is a subset of semi-active schedules. However, the set of non-delay schedules is much smaller than the active schedules. On a non-delay schedule, no machine will be idle until it can start processing (Pinedo, 2008). A set of non-delayed schedules is a subset of active programs, but it is possible to ignore the optimal solution. We suggest that a random number be generated in the interval [1, 0]. When this number is less than probability  $\beta$ , the solution is obtained without delay and using the ERT (Earliest Release Time) rule in the stages following the first process stage, while if the random number exceeds the probability of  $\beta$ , the initial phase sequence is used to assign the next stages of production. At the same time, the work can be carried out at an idle time if it does not delay any other work. The general structure of the proposed algorithm for generating active schedules is as follows:

**Step 1:** The matrix *X* is generated in size  $(n_i, n_k)$ .

**Step 2:** At the first stage (k = 1):

**Step 2-1:** The first column of the matrix X is sorted and base on it, the sequence of operations is determined (Matrix  $Y_1$ ).

**Step 2-2:** Based on the  $Y_1$  sequence, jobs are assigned to the first free machine and scheduled.

**Step 3:** From the second stage (k > 1):

**Step 3-1:** The random number b generated, if  $b > \beta$ , the solutions are based on the kth column of the matrix X and if  $b < \beta$  solutions are chosen based on the previous step matrix  $(Y_{k-1})$  and the sequence matrix of the kth stage is obtained  $(Y_k)$ .

**Step 3-2:** All idle intervals are calculated for each machine in stage k, equals  $[I_i, I_{i+1}]_m$ .

**Step 3-3:** Based on the sequence of  $Y_k$ , the selected job is assigned to the first interval  $[I_i, I_{i+1}]$  with the minimum start time,  $(max[I_i, r_{i+k-1}])$ , where  $r_{i+k-1}$  is the finishing time on the previous stage. If:  $p_i + max[I_i, r_{i+k-1}] \le I_{i+1}$ .

**Step 3-4:** If there is a job in  $Y_k$ , go back to step 2-2 otherwise, go back to step 3-1.

In figure 2, the scheduling and routing chart of vehicles is drawn for example in the previous section. According to Table 1, the production times for each job at each stage  $(p_{j,k})$  and according to Table 2, the transportation times of each job  $(t_{i,i'})$  are as follows. The machines are shown with the symbol  $M_{l,k}$ , which represents the lth machine of the kth production stage. As shown in the Gantt chart, the third job in the third stage begins with the presentation of the active solution before the fourth job, while in the initial sequence, the priority of the third job, after the fourth job, is given. The reason for this is that without other, any delay in other jobs with more priority (that is, the first, second, and fourth jobs in the third stage), processing of the third job could be started and finished in the vehicle idleness and before the fourth job entering to the next production stage, which is a feature of active solution representation.

Table1. Process	ing time v	alues (	$(p_{j,k})$ for	the ex	ample
Job j	1	2	3	4	5
Stage 1	1	2	2	3	4
Stage 2	4	2	2	4	4
Stage 3	3	1	1	3	1

**Table2.** Processing time values  $(t_{ii'})$  for the example Location 

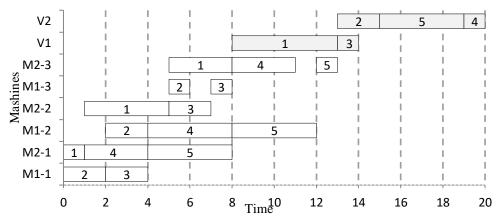


Fig2. Gantt chart of scheduling and vehicles routing

According to this solution, the jobs are processed on machines. For example, the second machine of the third stage  $(M_{2.3})$  finishes the production process of jobs 1, 4, and 5 at time 8, 11, and 13, respectively. In this way, jobs pass the stages in sequence. For example, job 3 starts its process on the first stage at time 2 until 4, then it stars the second stage at time 5 until 7 and it goes to the third stage at time 7 until 8. Finally, it is delivered to the customer at time 14 (after waiting for 5 min to release vehicle 1).

jobs 1 and 3 are delivered at the time of 13 and 14, respectively, by the first vehicle, and goods 2, 5, and 4 are delivered by the second vehicle at the times of 15, 19, and 20 respectively.

#### 4-5-Initial population generation

Research on scheduling problems includes many articles that emphasize the impressive impact of a good initial solution on metaheuristic algorithms. Today, it is hardly possible to make hyper business at least in scheduling, with the initial solution given only by a random solution. Usually, the results of randomized solutions are converted into better solutions by methods. Therefore, it is necessary to carefully consider the choice of the conversion procedure for the initial solution to achieve a high level of performance and competitiveness in scheduling problems.

We can use heuristic algorithms to generate the initial population. Depending on the due dates, methods for sorting and sequencing such as due date rules can be used. Kim (1993) arranged jobs according to the EDD rule, respectively, their non-descending order of delivery. Also, another way of sorting jobs has been proposed, in which the job is arranged in an orderly manner of delivery dates, that is, from the Longest Due Date (LDD). Also, Kaweegitbundit (2012) conducted a review of the various dispatching rules for HFS problems, to minimize the completion time and minimize the total tardiness. In this paper, the two-stage HFS with the same parallel machines is considered the performance of the dispatching rules is reviewed. He showed that the EDD is better than other rules to minimize the total tardiness as an objective.

In this paper, the initial population is constructed by the general rule of EDD, based on computational experiments. Given the parameter of the EDD method, we use a complete sequence of other jobs. In other words, a random number is generated in the interval [1, 0]. When this number is less than  $\gamma$  probability, the initial sequence would be created using the EDD rule, while if the random number is greater than probability, the initial sequence would be generated randomly. The final sequence for the particle is arranged so that it can be in the particle population. We repeat this method for all possible jobs in the main permutations, such as the first work, to make the initial population. By doing so, population diversity is achieved. The pseudocode of the initial population is presented in figure 5.

Then, for each particle, a matrix with random numbers is generated in the interval [1, J + V - 1], and as explained in the representation section, the initial solution is turned on the routing section, and the customers' allocation to the paths is done. In this way, the initial solution of the algorithm is created for the first particles.

#### 4-6-Particles updates

In this paper, a discrete PSO algorithm is used, which includes a local search based on the mutation operator that operates using hybrid neighborhoods and crossover operators, which is first presented in Pan et al. (2008). Considering the importance of the initial solution in the solving method and considering the objective function of the problem, which includes minimizing the total tardiness, the initial population will be created by a combination of constructive heuristic methods. After updating the particle position and creating the next population, the comparison of the particle matching function is performed and the best individual and global solution will be updated.

Since a solution is indicated by the jobs permutation as (1, 2, ..., n), the particle position can be updated according to the following constraint (Penn et al., 2008).

$$X_i^t = c_2 \otimes F_3 (c_1 \otimes F_2 (\omega \otimes F_1 (X_i^{t-1}), P_i^{t-1}), G^{t-1})$$
 (25)

Note that  $X_i^t$  is the particle position  $P_i^t$  is the best individual solution to the particle and  $G^t$  is the best

- global solution. The updated equation contains three components

  1. The first component  $A_i^{t} = \omega \otimes F_1(X_i^{t-1})$  represents the particle velocity and  $F_1$  is the mutation operator with probability  $\omega$ .
- 2. The second component B<sub>i</sub><sup>t</sup> = c<sub>1</sub> Ø F<sub>2</sub> (A<sub>i</sub><sup>t</sup>, P<sub>i</sub><sup>t-1</sup>) is related to the individual section of the particle and F<sub>2</sub> is the crossover operator with probability c<sub>1</sub>.
  3. The third component C<sub>i</sub><sup>t</sup> = c<sub>2</sub> Ø F<sub>3</sub> (B<sub>i</sub><sup>t</sup>, G<sub>i</sub><sup>t-1</sup>) is related to the global section of the particle and F<sub>3</sub> is the crossover operator with probability c<sub>2</sub>.
- is the crossover operator with probability  $c_2$ .

Various operators can be used for the algorithm; we will continue to describe these operators.

#### 4-6-1-Mutation operator

The mutation operator creates random variations in a particle to maintain the diversity of the population at a reasonable level. Ho et al. (2008) used an inversion mutation (Gen & Cheng, 1997) in a hybrid genetic algorithm to solve the VRP. Mirabi (2014) used an inversion mutation operator in a hybrid genetic algorithm to solve the FS scheduling problem. Niu et al. (2010) used a reverse mutation operator in the genetic algorithm to solve the HFS scheduling problem.

In the proposed algorithm, the inverse mutation is applied equally to one of the two parts of the solution vector. In this mutation, two positions are randomly selected, and then the section between the two positions is reversed. In this way, two random positions are selected in the solution and a random number is generated between 0 and 1. If the generated number is larger than the parameter  $\omega$  is larger, the job sequence is reversed between these two positions.

The following example shows the mutating process. Suppose that a problem has been defined with 7 jobs two vehicles and the following solution is chosen to perform a mutation. First, the selected particle in the chromosome will be copied to the new solution. Then two numbers between 1 and the number of jobs will be generated randomly in part one of the solution's vector and two numbers between 1 and the number of jobs and vehicles will be generated randomly in part two of the solution's vector. The sequence between these two positions is reversed and transmitted to the new solution in each part (figure 3). In the EAPG algorithm mutation operator runs on both parts of the solution vector simultaneously and separately.

Mutation candidate	3	5	2	6	1	4	7	3	5	7	1	6	4	9	8	2
Mutated solution	3	1	6	2	5	4	7	3	5	7	8	9	4	6	1	2

Fig3. Mutation operator

#### 4-6-2-Crossover operator

In this paper, the PMX crossover operator is used which is presented by Ahmadizar and Farahani (2012). A crossover operator is proposed in which random numbers are used to determine the selected parent to create the current child's chromosome. A random number is generated for each job. If the value is less than  $c_1$ , the amount of the first chromosome will be copied to the new chromosome, whose random number generated by them is more than  $c_1$ , will be transferred to the new chromosome in the second chromosome order. This operator is used for the first time in the PSO algorithm for the production flow and the result is a child for two parents, because one of the parents in the PSO will be the personal best or global best solution, and only the good genes are transmitted to the child from each particle.

- **Step 1.** Copy the parent 1 into the child's chromosome.
- **Step 2.** For k = 1 job, do the following:
  - **2-1.** Create a random number  $c_1$  between 0 and 1.
  - **2-2.** If  $C_1 > c_1$ , then substitute those jobs in the child's chromosomes by their order in the second parent.

The example below shows the crossover process. Suppose a problem with 9 jobs is defined and the following two solutions are selected for the crossover (figure 4).

The child's chromosome has been copied from the beginning of the first parent. Then for each job, a random number is generated between 0 and 1. Assume that for all cases, random numbers are less than 0.7, except for the fifth, seventh, and ninth editions of random numbers greater than 0.7. Therefore, the sequence of this job should be replaced. Fifth, seventh, and ninth steps in the second parent with a sequence of 7-6-9 are performed, thus they are transmitted to the child with the same sequence (figure 5). In the EAPG algorithm crossover operator runs on both parts of the solution vector simultaneously and separately.

Parent 1	3	5	2	6	1	4	7	3	5	7	1	6	4	9	8	2
Parent 2	1	3	6	2	7	5	4	2	6	3	7	1	9	8	5	4

Fig4. Crossover operator

The child's chromosome has been copied from the beginning of the first parent. Then for each position, a random number is generated between 0 and 1. Assume that for all cases, random numbers are less than 0.7, except for the fifth, sixth, and seventh positions in part one and the fifth, first, and eighth positions in part two that random numbers greater than 0.7. Therefore, the sequence of this position should be replaced. Fifth, sixth, and seventh position in part one, in the second parent with a sequence of 6-7-5 and fifth, first and eighth position in part two, in the second parent with a sequence of 1-8-5 are performed, thus they are transmitted to the child with the same sequence (figure 5). In the EAPG algorithm crossover operator runs on both parts of the solution vector simultaneously and separately.

Offspring	3	6	2	7	1	4	5	3	1	7	8	6	4	9	5	2

Fig 5. Offspring in crossover operator

# 4-7-Stop condition

The criterion that is considered to stop the implementation of the PSO algorithm is the maximum iteration number.

# **5-Computational results**

In this section, to evaluate the performance of the mathematical model and the proposed algorithm, we compare the computational results obtained from the CPLEX solver with the algorithm solutions in the

Matlab environment (the proposed model is nonlinear; we linearize the proposed model by using CPLEX solver). The comparison criterion is the average of the solutions in ten iterations of each problem with the solution given by the metaheuristic algorithm (and in the case of small sizes, the solution obtained by solving with CPLEX solver). To perform the comparison, the problem solving with the proposed algorithm, along with the PSO metaheuristic algorithm, is performed in the Matlab software environment with a personal computer with a RAM of 6 GB and a 2.20 GHz processor.

#### 5-1-Data generation

To data generation, the parameters of the problem are defined as the number of jobs, the number of machines, the number of stages, processing times and fixed costs, and the number of vehicles, the capacity of vehicles, the cost and time between customers and the due date and the penalty for tardiness. Each order is identified. To determine some of the parameter's value, Chao and Qing (2008) have been helpful. These values are shown in table 3. A total of 60 problems have been created, which are available at http://web.ntust.edu.tw/~ie/index.html as well as the OR library at http://mscmga.ms.ic.ac.uk/info.html. For processing times, numbers are generated on standard problems are used. In this article, the locations are created by randomly generated latitudes and locations, and for our study; we can calculate the distance points of the points from these locations. Because of the calculation of the distance points from their geographic coordinates, they guarantee the Triangle inequality, so it is closer to the reality than the randomized method of the distance matrix. The data on the problem is described in table 3.

Table3. Different levels of identified factors in the problem

row	parameter	Parameter notation	Parameter range	references
1	Stage numbers	$n_k$	2-5-8-10	
2	Machines numbers	$n_m$	2-3-4-5	Oguz et al (2004)
3	Job numbers	$n_{j}$	5-10-20-50-100	
4	Processing time	$p_{j,k}$	U(1,100)	
5	Due date	$d_j$	)/2((1 - TAR $\pm$ RDD/2) $\sum_{j}\sum_{k}P_{j,k}$	Ahmadizar and Farhadi (2015)
6	Travel time between two customer's location	$t_{i,i'}$	$x_{Depot} = y_{Depot} = U (20, 50)$ $x_{i} = U (0, 2 \times x_{Depot})$ $y_{i} = U (0, 2 \times y_{Depot})$ $dis_{i,i'} = \sqrt{(x_{i'} - x_{i})^{2} + (y_{i'} - y_{i})^{2}}$ $t_{i,i'} = dis_{i,i'}$	
7	Travel cost between two customer's location	$c_{i,i}$ '	$unitcost = U(50,200)$ $= unitcost * dis_{i,i'}c_{i,i'}$	Ramezanian et al (2017)
8	Vehicle's capacity	$cap_v$	Small size problem 4 Medium size problem 10 Large size problem 15	
9	Vehicle's fix cost	fc	<i>U</i> (150,200)	
10	Tardiness penalty	$pi_i$	U(5,15)	Generated data

It is noteworthy that the algorithm parameters have been adjusted to match the dimensions of the problem in different situations, but the algorithm will also change in these situations. For this reason, a constant structure of the algorithm's characteristics, such as the number of iterations and the number of particles, has been used to solve the problem.

## 5-2-Adjusting parameters

The various values of the existing parameters have a significant effect on the quality of the solutions obtained from the particle swarm algorithm. At this point, we conducted a series of experiments to determine the parameters of the proposed algorithm. The parameter  $Iter_{max}$  is the number of iterations of the algorithm, which is proportional to the number of jobs and stages of production and is equal to  $10(n \times k)$ . Also, the parameter  $n_{pop}$  is the number of populations corresponding to the number of jobs equal to 3n. The parameters  $\beta$  and  $\gamma$  are respectively the probability of determining the initial solution according to the ERT and EDD rules and are determined in the range (0.3, 0.8). The probability  $\omega$  is determined for applying the mutation operator on each part (scheduling part and vehicle routing part) of the solution vector of each particle in the range (0.2, 0.8) and probabilities  $c_1$  and  $c_2$  for applying crossover operators on each part (scheduling part and vehicle routing part) of the solution vector of each particle and in the range (0.7, 0.9). To obtain the best combination of parameters for the proposed algorithm, different combinations of parameters are determined. After solving different problems with different sizes, the parameters for the algorithm are obtained as follows. The best combination of the parameters mentioned in table 4 is given.

**Table4.** Adjusting parameters

Value	Parameter	Notation
0.5	mutation probability	ω
0.7	crossover probability with the best personal solution	$c_1$
0.9	crossover probability with the best global solution	$c_2$
0.5	EDD probability	γ
0.3	ERT probability	β

#### 5-3- Comparing the results

Since we have not found a similar study in the literature, the PSO algorithm has been used to evaluate the efficiency of the proposed algorithm. The comparison criterion is the mean of the solutions in ten iterations of each problem with the solution given by the metaheuristic algorithm (and in the case of small sizes, the solution obtained by solving with CPLEX solver).

The selected index for comparison is the relative deviation (RPDs) of the average of solutions of these iterations from the lowest obtained of algorithms. The deviation from the lowest solution is obtained from relation (30). In this equation, the lowest solution is shown by Min and the algorithm solution is shown by Alg.

$$RPD = \frac{Alg - Min}{Min} \tag{26}$$

In this section, the comparison between the efficiency of the proposed heuristic algorithm with metaheuristic algorithms in literature including GA and PSO is presented in Tables 5 and 6. For each dimension of the problem, the proposed PSO algorithm, which we display with EAPG (EDD-ERT-Active-PSO-GA), is recorded with the above characteristics, and its results are recorded. This algorithm searches in an active solution space and uses the EDD and ERT rules as a possible initial representation of the initial solution. In the next populations, the particle position update also will be done with mutation and crossover operators as described in sections 4-4-1 and 4-4-2. A simpler version of the algorithm is also used to examine the components of the algorithm separately in the analysis of the results, which we

call the EDD-PSO-GA (PSO3) algorithm, and search the entire space for the solution, and only in the Initial population of the EDD rule, randomly and in next populations, it uses mutation and crossover operators, but it does not use the ERT rule and search in the active environment. Two algorithms are also used to compare the components of the algorithm in the analysis of the results, which we call them the PSO-GA (PSO2) algorithm that uses the only mutation and crossover operators and the Classic PSO (PSO1) algorithm.

Given that the problem is NP-Hard, with the increase in the problem dimensions, CPLEX solver is unable to solve the problem in an acceptable time; therefore, in larger problems, the solution CPLEX received over two hours in comparisons has been used and for each problem size the solution quality is measured by the mean difference from the best-found solution and optimal values are indicated in boldface. In this comparison, the proposed algorithm has been run for each instance with the number of iterations equal to the  $Iter_{max}$ , five times and the best one has been chosen, and then, in the time that this algorithm reaches its best, other algorithms have been run and the results are recorded. These algorithms have been run five times for each instance within every problem size, and then the minimum, average, and maximum of the objective function values obtained have been presented.

Table 5 shows the comparison between the solutions obtained by CPLEX solver for small size problems with the proposed algorithm in this paper, the EAPG algorithm, and the PSO algorithm for different modes (number of steps × number of jobs) are presented.

Scheduling is a problem at the operational level and needs to be resolved quickly. Given that the solution time of the proposed algorithm is much less than the exact solution time; Also, in solving problems for some problems even after the maximum time has not been a good solution, the use of this algorithm is time-saving. On the other hand, considering the difference between the exact solution and the solution of the proposed algorithm in small instances, the use of this algorithm is appropriate in terms of solution accuracy. Given the low error rate of the algorithm, it can be inferred that the amount of error is not large for medium and large problems that do not have an exact solution; So this algorithm can be used to solve large problems.

**Table5.** Comparison between solutions of small-scale problems

		MI	LP		PSO1	1	ison betwe	PSO2			PSO3			EA	PG	
problem n-m-k	Instance	Quality	Time	Min	Ave	Max	Min	Ave	Max	Min	Ave	Мах	Min	Ave	Max	Time
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2
3-2-2	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1
	Ave	0	1.3	0	0	0	0	0	0	0	0	0	0	0	0	1
	1	0	21	0	0.01	0.07	0	0	0	0	0	0	0	0	0	3
5-2-2	2	0	19	0	0.02	0.08	0	0	0	0	0	0	0	0	0	2
	3	0	14	0.03	0.06	0.10	0	0	0	0	0	0	0	0	0	1
	Ave	0	18	0.01	0.03	0.08	0	0	0	0	0	0	0	0	0	2
	1	0	48	0.08	0.19	0.35	0	0	0	0	0	0	0	0	0	2
5-2-5	2	0	65	0.15	0.36	0.52	0	0	0	0	0	0	0	0	0	2
	3	0	95	0.04	0.07	0.11	0	0	0	0	0	0	0	0	0	3
	Ave	0	69.3	0.09	0.20	0.33	0	0	0	0	0	0	0	0	0	2
	1	0	2	0	0.01	0.04	0	0	0	0	0	0	0	0	0	4
5-2-8	2	0	8	0.24	0.41	0.68	0	0	0	0	0	0	0	0	0	2
	3	0	4	0.06	0.13	0.24	0	0	0	0	0	0	0	0	0	3
	Ave	0	4.6	0.10	0.18	0.32	0	0	0	0	0	0	0	0	0	3
	1	0	4	0.29	1.19	2.17	0	0	0	0	0	0	0	0	0	7
5-2-10	2	0	5	0.04	0.25	0.39	0	0	0	0	0	0	0	0	0	5
	3	0	5	0.07	0.29	0.45	0	0.01	0.03	0	0	0	0	0	0	5
	Ave	0	4.7	0.13	0.58	1.00	0	0	0.01	0	0	0	0	0	0	5.7
	1	0	6	0.12	0.15	0.18	0	0.06	0.27	0	0	0	0	0	0	14
10-3-2	2	0	5	0.13	0.21	0.27	0	0.04	0.06	0	0	0	0	0	0	14
	3	0	5	0.16	0.22	0.28	0	0	0.02	0	0	0.01	0	0	0	15
	Ave	0	5.3	0.14	0.19	0.24	0	0.04	0.12	0	0	0	0	0	0	14.3
	1	0	4	0.05	0.11	0.15	0	0	0	0	0	0	0	0	0	31
10-3-5	2 3	0	3	0.57	1.00	1.92	0.09	0.11	0.19	0	0.06	0.09	0	0	0	32
		0.2 0.06	5	0.20 0.27	0.23	0.27 0.78	0.05 0.05	0.06 0.06	0.08 0.09	0	0 0.02	0 0.03	0	0	0	31
	Ave		4		0.45					0			0	0	0	31.3
	1	0	2	0.92	2.08	3.14	0.11	0.23	0.29	0.01	0.19	0.22	0	0	0	48
10-3-8	2 3	0	3	0.50 0.42	1.59 0.56	3.32 0.65	0	0.05	0.06 0.08	0	<b>0</b> 0.02	<b>0</b> 0.06	0	0	0	49 52
		0	3 2.7	0.42	1.41	2.37	<b>0</b> 0.04	0.03		0	0.02	0.06	<b>0</b> 0	0	0	52 49.7
	Ave								0.14							
	1 2	0	4	0.50	0.68	0.88	0.02	0.08	0.12	0.02	0.04	0.07	0	0	0	42
10-3-10	3	0.3	3	0.32	0.58	1.09	0.07 <b>0</b>	0.09	0.13	0	0.07	0.11	0 <b>0</b>			40
		<b>0</b> 0.1	3.3	1.56 0.79	1.87 1.04	2.40		0.01	0.02	0.01	<b>0</b> 0.04	0		0.01	0.02	41
A 110m2 22	Ave					1.45	0.03	0.06	0.09			0.06	0	0	0	40
Average		0.02	12.58	0.24	0.45	0.73	0.01	0.03	0.05	0	0.02	0.02	0	0	0	16.6

From table 5, it can be observed that the EAPG obtains much better solutions in a shorter or comparable CPU time than other algorithms. Also, when comparing the best results achieved by the EAPG with the results of the three other PSO algorithms, the EAPG is superior to PSO1, PSO2, and PSO3. Table 6 shows the comparison between the metaheuristic algorithms as mentioned above (PSO1, PSO2, PSO3, and EAPG) for large-scale problems. To have a suitable comparison, the time limit of each run of PSO1 and PSO2 as well as PSO3 is set equal to the EAPG.

**Table6.** Comparison between solutions for large-scale problems

Problem Firmity					1 abi	<b>eo.</b> Co.	mpariso	n betwee	II SOIUI	.10118 101	r large-sca	ne proc	nems				
		9.	M	ILP		PSO1			PSO2			PSO3			EA	PG	
20         0         110         0.84         0.88         0.90         0.33         0.40         0.04         0.03         0.04         0.03         0.04         0.03         0.04         0.05         0.04         0.03         0.04         0.05         0.04         0.05         0.04         0.05         0.04         0.05         0.04         0.05         0.04         0.05         0.04         0.05         0.04         0.05         0.04         0.05         0.04         0.04         0.04         0.04         0.04         0.04         0.04         0.04         0.04         0.04	-	Instanc	Quality	Time	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max	Time
1		1	0	120	0.84	1.05	1.26	0.01	0.04	0.67	0.02	0.04	0.11	0	0	0.01	47
No.   No.		2	0	110	0.84	0.88	0.90	0.33	0.42	0.49	0	0.08	0.12	0	0.01	0.02	43
1	20-4-2	3	0	395	0.71	0.78	0.83	0.17	0.19	0.20	0.01	0.14	0.18	0	0	0	42
Parish   P		Ave	0	208	0.79	0.90	1.03	0.17	0.22	0.45	0.01	0.09	0.14	0	0	0.01	44
1		1	0	162	0.26	0.45	0.57	0.80	1.01	1.67	0	0.02	0.04	0	0.03	0.06	117
1	20.4.5	2	0	138	0.54	0.63	0.71	0.31	0.39	0.43	0.16	0.18	0.21	0.01	0.03	0.06	110
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	20-4-5	3	0.1	144	0.94	1.19	1.51	0.21	0.26	0.31	0.05	0.13	0.20	0	0.01	0.02	110
20.4-8.8 $2$ $0$ $818$ $0.51$ $0.57$ $0.69$ $0.25$ $0.27$ $0.23$ $0.27$ $0.09$ $0.09$ $0.16$ $0.17$ $0.09$ $0.00$ $0.00$ $0.00$ $0.00$		Ave	0.03	148	0.58	0.76	0.93	0.44	0.55	0.80	0.07	0.11	0.15	0	0.02	0.04	112.3
1		1	0.4	713	0.46	0.56	0.64	0.53	0.59	0.64	0.38	0.45	0.64	0	0.02	0.05	150
No.   No.	20.40	2	0	818	0.51	0.57	0.69	0.25	0.26	0.27	0.23	0.25	0.27	0	0.09	0.16	170
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	20-4-8	3	0	738	0.43	0.60	0.80	0.79	0.86	0.89	0.74	0.78	0.86	0	0.04	0.06	118
$\begin{array}{l l l l l l l l l l l l l l l l l l l $		Ave	0.13	756	0.47	0.58	0.71	0.52	0.57	0.60	0.45	0.49	0.59	0	0.05	0.09	146
No.   No.	-	1	0	1240	0.81	0.92	1.15	0.62	0.76	0.87	0.45	0.49	0.55	0	0.11	0.13	164
No.   No.	20.4.10	2	0	1740	0.64	0.74	1.02	0.49	0.58	0.67	0.22	0.31	0.38	0.06	0.12	0.15	163
1   0.74   7200   0.46   0.54   0.69   0.09   0.12   0.19   0.02   0.10   0.12   0   0.02   0.05   131	20-4-10	3	0.8	1950	0.61	0.73	0.95	0.31	0.40	0.52	0.25	0.28	0.33	0	0.03	0.09	165
50-5-2         2         0.38         7200         0.64         0.76         0.89         0.22         0.35         0.41         0.02         0.04         0.06         0         0.01         0.06         159           50-5-2         3         1.69         7200         0.73         0.87         0.92         0.53         0.57         0.59         0.05         0.06         0.09         0         0.01         0.04         172           Ave         0.93         7200         0.61         0.72         0.83         0.28         0.35         0.40         0.03         0.06         0.09         0         0.01         0.05         154           50-5-5         4         0.93         7200         0.90         1.09         1.45         0.08         0.13         0.19         0.06         0.09         0.12         0         0.02         0.07         411           50-5-5         3         3.43         7200         1.17         2.01         2.14         0.16         0.23         0.30         0.21         0.35         0.44         0         0.02         0.15         544           40-5-5-8         3         3.44         7200         1.13		Ave	0.26	1643	0.68	0.79	1.04	0.47	0.58	0.68	0.31	0.36	0.42	0.02	0.09	0.12	164
50-5-2         3         1.69         7200         0.73         0.87         0.92         0.53         0.57         0.59         0.05         0.06         0.09         0         0.01         0.04         172           Ave         0.93         7200         0.61         0.72         0.83         0.28         0.35         0.40         0.03         0.06         0.09         0         0.01         0.05         154           4         2.39         7200         0.90         1.09         1.45         0.08         0.13         0.19         0.06         0.09         0.12         0         0.02         0.07         411           50-5-5         2         4.49         7200         1.54         2.41         3.41         0.09         0.15         0.20         0.11         0.25         0.35         0         0.05         0.03         481           50-5-5         3         3.43         7200         1.17         2.01         2.14         0.16         0.23         0.30         0.21         0.35         0.44         0         0.02         0.15         544           Ave         3.44         7200         1.13         1.57         2.47 <t< td=""><td></td><td>1</td><td>0.74</td><td>7200</td><td>0.46</td><td>0.54</td><td>0.69</td><td>0.09</td><td>0.12</td><td>0.19</td><td>0.02</td><td>0.10</td><td>0.12</td><td>0</td><td>0.02</td><td>0.05</td><td>131</td></t<>		1	0.74	7200	0.46	0.54	0.69	0.09	0.12	0.19	0.02	0.10	0.12	0	0.02	0.05	131
3   1.69   7200   0.73   0.87   0.92   0.53   0.57   0.59   0.05   0.06   0.09   0   0.01   0.04   172	50.5.2	2	0.38	7200	0.64	0.76	0.89	0.22	0.35	0.41	0.02	0.04	0.06	0	0.01	0.06	159
1 2.39 7200 0.90 1.09 1.45 0.08 0.13 0.19 0.06 0.09 0.12 0 0.02 0.07 411  50-5-5 2 4.49 7200 1.54 2.41 3.41 0.09 0.15 0.20 0.11 0.25 0.35 0 0.05 0.03 481  Ave 3.44 7200 1.20 1.64 2.23 0.11 0.17 0.23 0.13 0.23 0.30 0 0.00 0.03 0.08 479  1 2.51 7200 1.13 1.57 2.47 0.31 0.45 0.58 0.05 0.11 0.18 0 0.01 0.07 724  2 2.62 7200 0.91 1.05 1.27 0.29 0.31 0.23 0.11 0.19 0.23 0 0.00 0.01 0.03 716  3 3.89 7200 0.72 0.93 1.07 0.16 0.20 0.23 0.38 0.10 0.17 0.23 0.00 0 0.02 0.05 722  Ave 3.06 7200 0.92 1.18 1.60 0.25 0.32 0.38 0.10 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.88 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0.25 0 0.05 0.11 133	30-3-2	3	1.69	7200	0.73	0.87	0.92	0.53	0.57	0.59	0.05	0.06	0.09	0	0.01	0.04	172
50-5-5         2         4.49         7200         1.54         2.41         3.41         0.09         0.15         0.20         0.11         0.25         0.35         0         0.05         0.03         481           50-5-5         3         3.43         7200         1.17         2.01         2.14         0.16         0.23         0.30         0.21         0.35         0.44         0         0.02         0.15         544           Ave         3.44         7200         1.20         1.64         2.23         0.11         0.17         0.23         0.30         0         0.03         0.08         479           50-5-8         3         3.44         7200         1.13         1.57         2.47         0.31         0.45         0.58         0.05         0.11         0.18         0         0.01         0.07         724           50-5-8         2         2.62         7200         0.91         1.05         1.27         0.29         0.31         0.23         0.11         0.18         0         0.01         0.03         716           Ave         3.06         7200         0.92         1.18         1.60         0.25         0.32		Ave	0.93	7200	0.61	0.72	0.83	0.28	0.35	0.40	0.03	0.06	0.09	0	0.01	0.05	154
50-5-5  3 3.43 7200 1.17 2.01 2.14 0.16 0.23 0.30 0.21 0.35 0.44 0 0.002 0.15 544  Ave 3.44 7200 1.20 1.64 2.23 0.11 0.17 0.23 0.13 0.23 0.30 0 0.03 0.08 479  1 2.51 7200 1.13 1.57 2.47 0.31 0.45 0.58 0.05 0.11 0.18 0 0.01 0.07 724  2 2.62 7200 0.91 1.05 1.27 0.29 0.31 0.23 0.11 0.19 0.23 0 0.01 0.03 716  3 3.89 7200 0.72 0.93 1.07 0.16 0.20 0.23 0.14 0.21 0.27 0 0.04 0.06 725  Ave 3.06 7200 0.92 1.18 1.60 0.25 0.32 0.38 0.10 0.17 0.23 0 0.02 0.05 722  1 3.95 7200 1.07 1.10 1.47 0.07 0.12 0.20 0.11 0.14 0.20 0 0.03 0.05 386  2 3.88 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0.25 0 0.05 0.11 133	-	1	2.39	7200	0.90	1.09	1.45	0.08	0.13	0.19	0.06	0.09	0.12	0	0.02	0.07	411
3 3.43 7200 1.17 2.01 2.14 0.16 0.23 0.30 0.21 0.35 0.44 0 0.002 0.15 544  Ave 3.44 7200 1.20 1.64 2.23 0.11 0.17 0.23 0.13 0.23 0.30 0 0.03 0.08 479  1 2.51 7200 1.13 1.57 2.47 0.31 0.45 0.58 0.05 0.11 0.18 0 0.01 0.07 724  2 2.62 7200 0.91 1.05 1.27 0.29 0.31 0.23 0.11 0.19 0.23 0 0.01 0.03 716  3 3.89 7200 0.72 0.93 1.07 0.16 0.20 0.23 0.14 0.21 0.27 0 0.04 0.06 725  Ave 3.06 7200 0.92 1.18 1.60 0.25 0.32 0.38 0.10 0.17 0.23 0 0.00 0.02 0.05 722  1 3.95 7200 1.07 1.10 1.47 0.07 0.12 0.20 0.11 0.14 0.20 0 0.03 0.05 386  50-5-10 3 3.89 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0 0.05 0.11 133	50.5.5	2	4.49	7200	1.54	2.41	3.41	0.09	0.15	0.20	0.11	0.25	0.35	0	0.05	0.03	481
1 2.51 7200 1.13 1.57 2.47 0.31 0.45 0.58 0.05 0.11 0.18 0 0.01 0.07 724  2 2.62 7200 0.91 1.05 1.27 0.29 0.31 0.23 0.11 0.19 0.23 0 0.01 0.03 716  3 3.89 7200 0.72 0.93 1.07 0.16 0.20 0.23 0.14 0.21 0.27 0 0.04 0.06 725  Ave 3.06 7200 0.92 1.18 1.60 0.25 0.32 0.38 0.10 0.17 0.23 0 0.02 0.05 722  1 3.95 7200 1.07 1.10 1.47 0.07 0.12 0.20 0.11 0.14 0.20 0 0.03 0.05 386  2 3.88 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0.25 0 0.05 0.11 133	30-3-3	3	3.43	7200	1.17	2.01	2.14	0.16	0.23	0.30	0.21	0.35	0.44	0	0.02	0.15	544
50-5-8 2 2.62 7200 0.91 1.05 1.27 0.29 0.31 0.23 0.11 0.19 0.23 0 0.01 0.03 716 3 3.89 7200 0.72 0.93 1.07 0.16 0.20 0.23 0.14 0.21 0.27 0 0.04 0.06 725  Ave 3.06 7200 0.92 1.18 1.60 0.25 0.32 0.38 0.10 0.17 0.23 0 0.02 0.05 722  1 3.95 7200 1.07 1.10 1.47 0.07 0.12 0.20 0.11 0.14 0.20 0 0.03 0.05 386  2 3.88 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0 0.05 0.11 133		Ave	3.44	7200	1.20	1.64	2.23	0.11	0.17	0.23	0.13	0.23	0.30	0	0.03	0.08	479
50-5-8 3 3.89 7200 0.72 0.93 1.07 0.16 0.20 0.23 0.14 0.21 0.27 0 0.04 0.06 725  Ave 3.06 7200 0.92 1.18 1.60 0.25 0.32 0.38 0.10 0.17 0.23 0 0.02 0.05 722  1 3.95 7200 1.07 1.10 1.47 0.07 0.12 0.20 0.11 0.14 0.20 0 0.03 0.05 386  2 3.88 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0 0.05 0.11 133		1	2.51	7200	1.13	1.57	2.47	0.31	0.45	0.58	0.05	0.11	0.18	0	0.01	0.07	724
3 3.89 7200 0.72 0.93 1.07 0.16 0.20 0.23 0.14 0.21 0.27 0 0.04 0.06 725  Ave 3.06 7200 0.92 1.18 1.60 0.25 0.32 0.38 0.10 0.17 0.23 0 0.02 0.05 722  1 3.95 7200 1.07 1.10 1.47 0.07 0.12 0.20 0.11 0.14 0.20 0 0.03 0.05 386  2 3.88 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147  50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0 0.05 0.11 133	50.5.9	2	2.62	7200	0.91	1.05	1.27	0.29	0.31	0.23	0.11	0.19	0.23	0	0.01	0.03	716
1 3.95 7200 1.07 1.10 1.47 0.07 0.12 0.20 0.11 0.14 0.20 0 0.03 0.05 386 2 3.88 7200 0.92 1.06 1.32 0.18 0.25 0.31 0.17 0.23 0.29 0 0.01 0.03 147 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0 0.05 0.11 133	30-3-8	3	3.89	7200	0.72	0.93	1.07	0.16	0.20	0.23	0.14	0.21	0.27	0	0.04	0.06	725
50-5-10     2     3.88     7200     0.92     1.06     1.32     0.18     0.25     0.31     0.17     0.23     0.29     0     0.01     0.03     147       3     3.97     7200     0.81     0.94     0.98     0.24     0.28     0.32     0.17     0.20     0.25     0     0.05     0.11     133		Ave	3.06	7200	0.92	1.18	1.60	0.25	0.32	0.38	0.10	0.17	0.23	0	0.02	0.05	722
50-5-10 3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0 0.05 0.11 133		1	3.95	7200	1.07	1.10	1.47	0.07	0.12	0.20	0.11	0.14	0.20	0	0.03	0.05	386
3 3.97 7200 0.81 0.94 0.98 0.24 0.28 0.32 0.17 0.20 0.25 0 0.05 0.11 133	50-5-10	2	3.88	7200	0.92	1.06	1.32	0.18	0.25	0.31	0.17	0.23	0.29	0	0.01	0.03	147
Ave 3.93 7200 0.93 1.03 1.25 0.16 0.22 0.28 0.15 0.19 0.25 0 0.03 0.06 222	50-5-10	3	3.97	7200	0.81	0.94	0.98	0.24	0.28	0.32	0.17	0.20	0.25	0	0.05	0.11	133
		Ave	3.93	7200	0.93	1.03	1.25	0.16	0.22	0.28	0.15	0.19	0.25	0	0.03	0.06	222

		MI	LP		PSO1			PSO2			PSO3			EAI	PG	
problem n-m-k	Instance	Quality	Time	Min	Ave	Мах	Min	Ave	Мах	Min	Ave	Мах	Min	Ave	Max	Time
	1	5.15	7200	1.02	1.09	1.21	0.39	0.46	0.51	0.18	0.20	0.24	0	0.03	0.06	880
100.5.2	2	3.28	7200	0.85	0.89	0.96	0.55	0.59	0.65	0.18	0.22	0.28	0	0.03	0.06	872
100-5-2	3	4.37	7200	0.88	0.91	0.99	0.41	0.52	0.60	0.17	0.21	0.26	0	0.05	0.07	885
	Ave	4.27	7200	0.92	0.96	1.05	0.45	0.52	0.59	0.18	0.21	0.26	0	0.04	0.06	879
	1	4.51	7200	0.95	1.02	1.06	0.62	0.71	0.82	0.15	0.23	0.28	0	0.05	0.08	935
100 5 5	2	6.62	7200	1.12	1.15	1.21	0.31	0.35	0.38	0.08	0.13	0.18	0	0.03	0.08	1250
100-5-5	3	4.89	7200	0.92	1.14	1.23	0.30	0.33	0.37	0.13	0.15	0.17	0	0.07	0.09	1164
	Ave	5.34	7200	1.00	1.10	1.16	0.41	0.46	0.52	0.12	0.17	0.21	0	0.05	0.08	1116
	1	4.89	7200	1.14	1.24	1.35	0.52	0.61	0.72	0.16	0.27	0.29	0	0.04	0.07	2670
100-5-8	2	6.23	7200	1.03	1.35	1.94	0.24	0.27	0.34	0.09	0.15	0.19	0	0.05	0.06	2556
100-5-8	3	6.01	7200	0.98	1.16	1.32	0.25	0.28	0.31	0.16	0.18	0.19	0	0.06	0.09	2280
	Ave	5.71	7200	1.05	1.25	1.53	0.33	0.39	0.46	0.14	0.20	0.22	0	0.05	0.07	2502
	1	5.80	7200	1.64	1.69	1.76	0.45	0.47	0.52	0.03	0.04	0.06	0	0.04	0.05	3120
100-5-10	2	5.16	7200	1.68	1.75	1.81	0.46	0.48	0.51	0.04	0.05	0.09	0	0.04	0.06	3185
100-5-10	3	6.33	7200	1.55	1.57	1.61	0.38	0.42	0.49	0.06	0.05	0.08	0	0.03	0.04	3325
	Ave	5.76	7200	1.62	1.67	1.73	0.43	0.45	0.51	0.04	0.05	0.08	0	0.04	0.05	3210
Average		2.76	5029	0.89	1.06	1.26	0.34	0.40	0.49	0.14	0.19	0.25	0	0.04	0.06	812

As expected, and it is clear from Tables 5 and 6, the use of the proposed MILP model is difficult when the problem size increases (in particular the number of jobs), and large sample problems cannot be solved optimally at a logical time. The efficiency of each algorithm is not better than the other. The reason for this difference in performance is that, when the number of stages in the flexible flow shop (FFS) production with parallel machines is low, algorithms can easily search the solution space and find the optimal solution. But when the number of stages of the problem increases, the solution space will increase, and algorithms will require more iteration and, in general, adjust their parameters to obtain the optimal solution. However, PSO3 and PSOGA algorithms showed better performance than PSO2 and PSO1 algorithms, and the performance of algorithms using the ERT and EDD rules is better than algorithms without using these rules. Finally, it is clear from the computational results that the proposed hybrid algorithm (EAPG) that searches in the active space and uses both practices has the best performance. In comparison between EAPG, PSO2, and PSO1, when the number of jobs increases to more than 50 jobs for some instances the worst results founded by PSO3 are better than the best results of PSO2, PSO1. Furthermore, in the comparison between EAPG and PSO3, their results are almost the same for the instance with up to 20 jobs but when the number of jobs increases to more than 20 jobs for some instances the worst results founded by EAPG are better than the best results of PSO3. This confirms the important role of local search to find a good solution.

From tables 5 and 6, it can then be seen that the EAPG is superior in 20 out of 21 problem sizes compared to PSO1, in 17 out of 21 problem sizes compared to PSO2, and in 16 out of 21 problem sizes compared to PSO3. On average, the EAPG that searches in the active space and uses both rules (ERT and EDD) outperforms these three algorithms (with an average of 0.841).

The comparison between the averages of time to reach the optimal solution by two presented algorithms for small size instances is shown in Fig 6. Also, the comparison between average values of algorithms for medium and large size instances with fixed iterations and fix time is shown in Fig 7 and 8.

Furthermore, in order to assess the significance of the differences between the results obtained by EAPG and the other metaheuristics, two-tailed paired t tests have been performed. The results of the t tests are given in Table 7 and the comparison between the EAPG algorithm and the three algorithms PSO1, PSO2, PSO3 is presented in terms of the deviation from the best solution. These 3 paired sample test performed at 95% confidence level with SPSS software. Null hypothesis  $(H_0)$  in these tests indicates no difference between algorithms and in alternative hypothesis  $(H_1)$ , there is a difference. Since the value of Sig (p-value) for all three paired sample tests are less than the error type 1 ( $\alpha = 0.05$ ) and the upper

bound and lower bound intervals do not contain zero, the null hypothesis is rejected in three tests and the results of the t tests indicate that, with strong statistical significance, EAPG has a better performance than PSO3 as well as PSO1 and PSO2.

**Table7.** Statistical significance of differences between the results for large-scale problems

				95% Confidence Inte	_	
		Mean	Std. Deviation	Lower	Upper	Sig. (2-tailed)
Pair 1	PSO1 - EAPG	1.00056	0.42998	0.85507	1.14604	0.000007
Pair 2	PSO2 - EAPG	0.33444	0.21497	0.26171	0.40718	0.00004
Pair 3	PSO3 - EAPG	0.12861	0.13964	0.08137	0.17586	0.0003

# 6-Conclusions and suggestions for future research

In this paper, the HFS scheduling problem with identical parallel machines and VRP is investigated, and for the first time, the problem of HFS scheduling has been integrated with the VRP to minimize the tardiness penalty and transportation. In the serial production environment, there are at the same time several parallel machines with the same speed and conditions. In the transport section, there are some vehicles equal to the number of jobs with limited capacity. A hybrid PSO algorithm with GA that searches in an active solution space is suggested to solve the integrated problem. The algorithm structure to solve the problem is specific and how to match the particle swarm optimization algorithm with the problem is shown. Due to the novelty of the literature studied, the problems are generated using random methods to examine the effectiveness and efficacy of the proposed method. To demonstrate the effectiveness of the proposed heuristic method, the heuristic algorithms presented in the literature are used. All algorithms are coded in the same software environment and its results are recorded. The computational results show that the proposed solution approach yields fairly good results in comparison with the current algorithms of the literature. The algorithm is capable to generate relatively good solutions for sample cases.

It is also suggested as a proposal for future research to examine the integrated FS production scheduling and routing for other modes of production environment such as heterogeneous machines or taking into account the setup time and the job sequence or taking into account the job inventory. The limited number of vehicles or the uses of vehicles at different speeds are among the suggestions for future research in this area.

#### References

Ahmadi Zar F., Hosseinabadi Farahani M., (2012), A novel hybrid genetic algorithm for the open shop scheduling problem, *International Journal of Advanced Manufacturing Technology*, 62, 775-787.

Ahmadizar F., Farhadi S., (2015). Single-machine batch delivery scheduling with job release dates, due windows and earliness, tardiness, holding and delivery costs, *Computers & Operations Research*, 53, 194-205.

Alba E., Dorronsoro B. (2004). Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms, Lecture Notes in Computer Science, (vol 3004). Springer, Berlin, Heidelberg.

Alvarez E., Díaz F., Osaba E., (2015), A multi-agent approach for dynamic production and distribution scheduling. *Int. J. Engineering Management and Economics*, 1, 1-20.

Amorim p., Belo-Filho M.A.F, Toledo F.M.B., Almeder C., Almada-Lobo B., (2013), Lot sizing versus batching in the production and distribution planning of perishable goods. *Int. J. Production Economics*, 146, 208–218.

Chen H.K., Hsueh C.F., Chang M.S., (2009), Production scheduling and vehicle routing with time windows for perishable food products, *Computers & Operations Research*. 36, 2009, 2311–2319.

Chen Z-L., (2010), Integrated Production and Outbound Distribution Scheduling. *Review and Extensions Operations Research*, 58(1), 130-148.

Chieu Ta Q., Billaut J.Ch., Bouquard J.L., (2015), Heuristic algorithms to minimize the total tardiness in a flow shop production and outbound distribution scheduling problem. *Presented at the 6th IESM Conference*, October 2015, Seville, Spain.

Choudhary K., Gautam G., Bharti N., Rathore V.S., (2019), Particle Swarm Optimization for Flexible Job Scheduling Problem with Mutation Strategy, *Computing and Network Sustainability*, 75, pp 497-503.

Diveev, A.I., Bobr O.V., (2017), Variational Genetic Algorithm for NP-hard Scheduling Problem Solution, *Procedia Computer Science*, 103, 52–58.

Eberhart R.C., Shi Y., (2000), Comparing inertia weights and constriction factors in particle swarm optimization, *In Proceedings of Congress on Evolutionary Computing*, 84–88.

Gen M., Cheng R., Genetic Algorithms and Engineering Design, Wiley, New York.

He S., Wu Q.H., Wen J.Y., Saunders J.R., Paton R.C., (2004), A particle swarm optimizer with passive congregation. *BioSystems*, 78, 135–147.

Ho W., Ho G.T.S., Ji P., Lau H.C.W., (2008), A hybrid genetic algorithm for the multi-depot vehicle routing problem, *Engineering Applications of Artificial Intelligence*, 21, 548–557.

Jamrus T., Chien Ch., (2018), Hybrid Particle Swarm Optimization Combined With Genetic Operators for Flexible Job-Shop Scheduling Under Uncertain Processing Time for Semiconductor Manufacturing, *IEEE Transactions on semiconductor manufacturing*, 31(1).

Johar F., Nordin S. Z., Potts C.N., (2016), Coordination of Production Scheduling and Vehicle Routing Problem with Due Dates. *American Institute of Physics Conference Proceedings*, 1750(1), 1–9.

Karimi N., Davoudpour H., (2015), A branch and bound method for solving multi-factory supply chain scheduling with batch delivery. *Expert Systems with Applications*, 42, 238–245.

Kaweegitbundit P., (2012), Evaluation Dispatching Rules for Two-Stage Hybrid Flow Shop Scheduling with Parallel Machines, *Applied Mechanics and Materials*, 152, 1487-1491.

Kennedy J., Eberhart R., (1997), A discrete binary version of the particle swarm algorithm, In Proceedings of the 1997 *IEEE international conference on systems*, 5, 4104–4108.

Kennedy J., Eberhart R., (1995), Particle swarm optimization, In Proceedings of the 1995 IEEE international conference on neural network, 4(4), 1942–1948.

Kennedy J., Eberhart R.C., Shi Y., (2001), Swarm Intelligence, (Morgan Kaufmann: CA).

Kim Y.D., (1993), A new branch and bound algorithm for minimizing mean tardiness in 2-machine flow shops. *Computers and Operations Research*, 20, 391–401.

Kumar R.S., Kondapaneni K., Dixit V., Goswami A., Thakur L.S., Tiwari M.K., (2015), Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. *Computers & Industrial Engineering*, 1-38.

Lacomme Ph., Moukrim A., Quilliot A., Vinot M., (2016), The Integrated Production and Transportation Scheduling Problem based on a GRASP×ELS resolution scheme. *International Federation of Automatic Control*, 49(12), 1466-1471.

Liao C.J., Tseng C.T., Luarn P., (2007), A discrete version of particle swarm optimization for flow shop scheduling problems, *Comput Oper Res*, 34, 3099–3111.

Low C., Chang C.M., Li R.K., and Huang C.L., (2014), Coordination of production scheduling and delivery problems with heterogeneous fleet. *International Journal of Production Economics*. 153(0), 139-148.

Mirabi M., (2014), A novel hybrid genetic algorithm to solve the sequence-dependent permutation flow-shop scheduling problem, *International Journal of Advanced Manufacturing Technology*, 71, 429–437.

Moons, S., Ramaekers, K., Caris, A., Arda, Y., (2017), Integrating production scheduling and vehicle routing decisions at the operational decision level: a review and discussion, *Computers & Industrial Engineering*, 104, 224-245.

Niu Q., Zhou F., Zhou T., (2010), Quantum Genetic Algorithm for Hybrid Flow Shop Scheduling Problems to Minimize Total Completion Time., *International Conference on Intelligent Computing for Sustainable Energy and Environment*, 21-29.

Oguz C., Ercan M., (2005), A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks, *J.Scheduling*, 8(4), 323-351.

Oguz C., Zinder Y., Do V.H., Janiak A., Lichtenstein M., (2004), Hybrid flow-shop scheduling problems with multiprocessor task systems, *European Journal of Operational Research*, 152(1), 115-131.

Pan Q.K., Tasgetiren M.F., Liang Y.CH., (2008), A discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem, *Computers & Operations Research*, 35, 2807 – 2839.

Ramezanian R., Mohammadi Sh., Cheraghalikhani A., (2017), Toward an integrated modeling approach for production and delivery operations in flow shop system: Trade-off between direct and routing delivery methods, *Journal of Manufacturing Systems*, 44, 79–92.

Rohmer S., Billaut J.CH., (2015), Production and outbound distribution scheduling: a two-agent approach. Presented at the 6th IESM Conference, Seville, Spain.

Serifoglu F.S., Ulusoy G., (2004), Multiprocessor task scheduling in multistage hybrid flow shops: a genetic algorithm approach, *J. Oper. Res. Soc*, 55, 504–512.

Shi Y., Eberhart R.C., (1998), A modified particle swarm optimizer, in *Proceedings of the IEEE Congress on Evolutionary Computation*, 1998, 69–73.

Tasgetiren M.F., Sevkli M., Liang Y.C., Gencyilmaz G., (2004), Particle swarm optimization algorithm for single machine total weighted tardiness problem, *in Proceeding of the IEEE Congress on Evolutionary Computation*, 1412–1419.