Vol. 9, No. 1, 2022, Pages 63-80

# JIEMS
Journal of Industrial Engineering and Management Studies

journal homepage: www.jiems.icms.ac.ir

## Hybrid flow shop scheduling and vehicle routing by considering holding costs

Fardin Ahmadizar*[1], Raheleh Moazami Goodarzi[1], Hiwa Farughi[1]

[1] Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran.

Received: Oct 2021-01/ Revised: Dec 2021-15/ Accepted: Mar 2022-09

**Abstract**

In this paper, a new model for hybrid flow shop scheduling is presented in which after the production is completed, each job is held in the warehouse until it is sent by the vehicle. Jobs are charged according to the storage time in the warehouse. Then they are delivered to customers by means of routing vehicles with limited and equal capacities. The problem's goal is finding an integrated schedule that minimizes the total costs, including transportation, holding, and tardiness costs. At first, a mixed-integer linear programming (MILP) model is presented for this problem. Due to the fact that the problem is NP-hard, a hybrid metaheuristic algorithm based on PSO algorithm and GA algorithm is suggested to solve the large-size instances. In this algorithm, genetic algorithm operators are used to update the particle swarm positions. The algorithm represents the initial solution by using dispatching rules. Also, some lemma and characteristics of the optimal solution are extracted as the dominance rules and are integrated with the proposed algorithm. Numerical studies with random problems have been performed to evaluate the effectiveness and efficiency of the suggested algorithm. According to the computational results, the algorithm performs well for large-scale instances and can generate relatively good solutions for the sample of investigated problems. On average, PGR performs better than the other three algorithms with an average of 0.883. To significantly evaluate the differences between the algorithms' solutions, statistical paired sample t-tests have been performed, and the results have been described for the paired algorithms.

**Keywords**: hybrid flow shop; vehicle routing; holding cost; dominance rules.

**Paper Type**: Original Research

### 1. Introduction

Planning is a process to make decisions in production and service systems such as manufacturing, transportation, and tourism (Taghizadehalvandi, and Kamisli Ozturk, 2019). Increasing competitions among companies are driving corporate executives to find new ways to reduce costs and maintain their competitive advantage while providing quality services for customers. The complexity of this global business environment makes scheduling and distribution problems critical and studying these problems can reduce cost, and increase the quality of a business's operations.

As the cost of holding and transporting goods has a prominent role in determining the actual price of products, one of the main targets of distribution companies is optimizing the distribution system. Distribution organizations are looking for efficient programs to provide better services for their customers to reduce their costs, and to satisfy customers. The practicality of problems related to the holding and transportation of goods and routing of vehicles along with the production schedule of goods, the complex structure of these problems has caught the researchers' attention in different fields such as compositional optimization and algorithm design and traffic management.

In the production of real goods, such as food and clothing, pharmaceutical and medical products, petrochemicals and chemicals, aircraft parts, and electronics, in addition to hybrid flow shop production environment, major costs, especially tardiness, holding, and transportation costs can be reduced through considering vehicle routing at the same time production is scheduled. So, it seems necessary to consider hybrid flow shop scheduling model integrated with vehicle routing along with the discussion of holding and transportation of goods. This problem can be modeled and solved and independent of a particular industry's characteristics and can be used for real-world industries.

Based on our knowledge and previous studies in this area, the integrated scheduling of the hybrid flow shop production and routing of limited-capacity vehicles that leads to the holding of inventory before distribution seems to have not been studied yet and it is necessary to study it.

In this paper, considering the effect of holding costs of ready-made goods that are waiting for other orders of the allocated vehicle, the holding cost has been investigated in the integrated model of hybrid flow shop scheduling, distribution, and vehicles routing. The holding cost of goods is determined based on their waiting time in a

*Corresponding Author: f.ahmadizar@uok.ac.ir


DOI: 10.22116/jiems.2022.307359.1461

warehouse after the production stage and before transportation by vehicles and it is minimized in the minimization objective function. In fact, the goal is minimizing the tardiness cost of delivering the orders to the customer, the holding cost of inventory, and fixed and variable transportation costs. To solve the problem of high cost, first, the MILP mathematical model based on the previously presented model is described. Then, according to the characteristics of the problem, the dominance rules are derived from the problem and are integrated with the proposed new mutation and crossover operators and dispatching rules with the hybrid particle swarm optimization algorithm. Different sections in this article are as follows: The second part includes the related works to scheduling and vehicle routing concerning holding the inventory of manufactured goods. In the third part, the definition of the problem and the mathematical model are mentioned along with a complete description and then in the fourth part, the dominance rules will be presented. In the fifth section, to solve the problem, a metaheuristic algorithm is suggested. Finally, in the seventh section, concluding comments and some suggestions for future research in this field are proposed.

## 2. Related works

A review of the most significant studies in the discussion of hybrid flow shop scheduling and vehicle routing and inventory holding costs is provided.

Mohammadi et al. (2020) expressed the need for integrated production and distribution planning in modern industries, especially custom products, where timely availability with a minimum total cost and meeting high customization requirements are challenging. To meet these challenges, they developed a flexible flow shop production schedule with time window vehicle routing. Their model had two objective functions: minimizing the cost of production planning and distribution as well as the weighted summation of the tardiness and earliness. Mousavi et al. (2020) presented a mathematical model in which the aim was to base orders on the existing air transportation capacities and, to prioritize them in a single machine production environment, and to minimize the entire chain costs. The goal includes tardiness, earliness, holding, and transportation costs.

In Rohmer and Billaut's (2015) research, the integrated production-distribution schedule including a manufacturer and logistics company has been investigated. Their problem is a flow shop production environment, and the goal of their model is to minimize inventory holding costs, vehicle costs, and tardiness penalties. Inventory costs in their model include holding costs for semi-finished and finished goods. This inventory holding time is obtained by subtracting the time each job is processed from the total time when it is in the production system. Vehicles have a fixed travel time. In this paper, vehicle routing is done to find the minimum value of the transportation and tardiness costs. The number of vehicles and their capacities is unlimited. Karimi and Davoudpour (2015) have considered a production, distribution schedule in a supply chain including manufacturers and suppliers. Production and delivery of the job are done in batches and each batch has a limited capacity. Each category is completed when the last job of that category is completed. The goal is to balance distribution costs with tardiness penalties. Also, a branch and bound method is suggested in which a lower bound and also, a heuristic method for determining the upper bound are introduced. In Karimi and Davoudpour's (2016) article, the integrated scheduling and transportation problem has been raised considering holding costs depend on the stages. The supply chain consists of several factories that are in series and aim to minimize the total costs of transportation and holding associated with the process. In the supply chain, various holding costs are imposed on the system from the beginning of the process to the end. Therefore, this study aims to find a balance between holding costs and delivery costs.

Ullrich (2013) studies the integrated production-distribution scheduling to fine the minimum tardiness cost. The problem is solved through a parallel machine scheduling of several jobs in accordance with the machine-dependent setup time and delivery of completed jobs. Holding costs have been used to find a balanced level between transportation costs, tardiness, and earliness costs. Moons et al. (2017) investigated production-distribution planning and considered vehicle routing in the distribution part. They reviewed the existing studies on the subject of integrated scheduling with vehicle routing and classified several articles. Hao et al. (2019) solved a distributed HFS. Jobs are first distributed between several factories and then production begins. Fu et al. (2019) considered a multi-objective stochastic model for distributed flow shop scheduling by minimizing tardiness and energy production and time consumption. The goal of their model was to reduce energy consumption along with reducing production costs in high-consumption industries through proper scheduling activities that were expensive at any time. Moazami et al. (2021) studied the integration between HFS scheduling and VRP and suggested a hybrid algorithm to solve it. Meinecke and Scholz Reiter (2014) and Scholz Reiter et al. (2011) also considered inventory between production and distribution stages, inventory holding costs along with fixed and variable transportation costs and tardiness penalties. In the study of Aghezzaf and Landeghem (2002), a hybrid flow shop production system with storage between two stages was presented. They tried to achieve the goal of optimizing production and holding costs in their model. Shabtay (2010) did the production and distribution scheduling to decrease tardiness and earliness cost, holding cost, and batch delivery costs. Agnetis et al. (2017) provided a production planning model taking into account batch holding and delivery costs. Izadi et al. (2020) presented scheduling and vehicle routing problem concerning outsourcing and inventory holding. Their model includes the parallel machines production environment and batch delivery according to the holding cost of the completed jobs. This study aimed to reduce the total costs of outsourcing, holding, distribution, production, delay. Wang et al. (2015) stated when an inventory holds before distribution stage, it can decrease production costs and increase the delivery speed. In their article, Izadi et al. (2018) examined the parallel machines scheduling and routing of vehicles considering batch delivery with the aim to gain an integrated schedule that minimizes the costs of setup, holding, distribution, and tardiness. Ayough and khorshidvand (2019) have proposed a new model to minimize the cost of cellular manufacturing system with limited number of cells and under uncertainty demands. They also have presented two algorithms based on Simulated Annealing (SA) and PSO. A brief review of related works is presented in Table 1.

**Table1. production and distribution scheduling problems and vehicle routing with regard to holding costs**

| row | Research | Production environment | Vehicle routing | Objective function | Inventory holding |
|---|---|---|---|---|---|
| 1 | Mohammadi et al. (2020) | Flexible Flow Shop | √ | Minimize production and transportation costs and weighted sum of delivery earliness and tardiness | - |
| 2 | Mousavi et al. (2020) | Single Machine | √ | Minimize transportation and holding costs and earliness and tardiness penalties | - |
| 3 | Rohmer and Billaut's (2015) | Flow Shop | √ | Minimize transportation and holding costs and tardiness penalties | Between production stages |
| 4 | Karimi and Davoudpour (2015) | Flow Shop | √ | Minimize transportation costs and tardiness penalties | - |
| 5 | Karimi and Davoudpour's (2016) | Flow Shop | √ | Minimize transportation and holding costs | Between production and distribution |
| 6 | Ullrich (2013) | Parallel Machines | √ | Minimize total tardiness | - |
| 7 | Hao et al. (2019) | Hybrid Flow Shop | - | minimize the maximum makespan | - |
| 8 | Fu et al. (2019) | Flow shop | - | minimize makespan and total energy consumption | - |
| 9 | Moazami et al. (2021) | Hybrid Flow Shop | √ | Minimize transportation costs and tardiness penalties | - |
| 10 | Scholz Reiter et al. (2011) | Flexible Flow Shop | √ | Minimize transportation and holding costs and tardiness penalties | Between production and distribution |
| 11 | Meinecke and Scholz Reiter (2014) | Flexible Cellular Manufacturing | √ | Minimize transportation and holding costs and tardiness penalties | Between production and distribution |
| 12 | Aghezzaf and Landeghem (2002) | Hybrid Flow Shop | - | Minimize transportation and production costs | Between production stages |
| 13 | Shabtay (2010) | Single Machine | √ | Minimize holding and batch delivery costs and earliness and tardiness penalties | Between production and distribution |
| 14 | Agnetis et al. (2017) | Parallel Machines | √ | Minimize transportation and batch delivery costs | Between production and distribution |
| 15 | Wang et al. (2015) | Single Machine | √ | Minimize setup, transportation and holding costs and earliness penalties | Between production and distribution |
| 16 | Izadi et al. (2018) | Parallel Machines | √ | Minimize setup, transportation and holding costs and tardiness penalties | Between production and distribution |
| 17 | Izadi et al. (2020) | Parallel Machines | √ | Minimize production, transportation, outsourcing and holding costs and tardiness penalties | Between production and distribution |
| 18 | Ayough and khorshidvand (2019) | Cellular Manufacturing | - | Minimize production, holding, maintenance and supply costs | Between production stages |
| 19 | current study | Hybrid Flow Shop | √ | Minimize transportation and holding costs and tardiness penalties | Between production and distribution |

According to the related works, it is obvious that the published articles on the integrating of flow shop production scheduling and vehicle routing has not considered inventory and holding costs and this area needs to be examined. This article presents a model with this feature. At each stage of production, identical parallel machines do the production. The jobs enter the stages in the prescribed order, and finally, the completed jobs enter the warehouse between the production and distribution stages and they are distributed by vehicles with limited and specific capacities. Any job will be stored in the warehouse and subject to the holding cost until all the jobs allocated to the vehicle are completed. The goal of modeling is to lower the overall costs of tardiness, holding, and transportation. Since the integrated problem, including scheduling an HFS and vehicle routing is very difficult (Moazami et al. 2021), if the dimensions of the problem increase, it is not possible to receive a global optimization during an acceptable time. So, to eradicate the problem presented in this paper, a hybrid PSO is presented to find an integrated solution.

## 3. Problem definition

The description of the problem and its assumptions are presented as follows:
Several products are produced in a multi-stage flow shop. Each stage involves several same parallel machines. Unlimited numbers of vehicles are available (by the number of orders). Vehicles have the same fixed and variable capacities and costs (they are homogeneous).
All machines and jobs are available at the beginning. A job does not go under processing on more than one machine simultaneously. The customers are located in different geographical places. Each customer orders one unit of product. The number of geographical points is equivalent to the customers' number plus depot node (factory location which is the origin and destination of vehicles $i = 1, N$)), so the customer who ordered job j is in node i, where $j = i - 1$. To reduce transportation costs and make more use of the vehicle's limited capacity, customer orders are shipped in batches. Each batch is a set of orders that includes one or more customers' orders and is carried by a vehicle and delivered to customers. Each vehicle goes on a tour that starts at the factory warehouse, then goes to the customers one by one to deliver their order, and finally returns to the factory warehouse. Vehicles have limited capacities and each job occupies one unit of vehicle capacity. Sending orders to customers has two types of transportation costs (fixed and variable), tardiness costs and holding costs. The fixed cost of transportation is the cost of using the vehicles and the variable cost of transportation is proportional to the time spent on transportation. Each customer has set a due date for their order. If we delay the delivery of the order to customers, the tardiness penalty must be paid in proportion to the duration of the tardiness. Since each vehicle contains several orders, if one of the orders is produced earlier than the others, it must wait in the warehouse before the production completion of other jobs. The holding of this product leads to the holding cost, which is proportionate to the storage time and the holding cost ratio of the product. Holding cost of a job is equal to the period between the release time of the vehicle which carries that job and that job's production completion time.

### 3.1. Mathematical Model

The mathematical formulation is as follows:

Indices

| | |
|---|---|
| $s$ | Number of stages ($s = 1,2, \dots, S$) |
| $m$ | Number of machines ($m = 1,2, \dots, M$) |
| $i$ | Number of customers ($i = 1,2, \dots, N$) |
| $j$ | Number of jobs ($j = 1,2, \dots, J$) |
| $v$ | Number of vehicles ($v = 1,2, \dots, V$) |

Parameters

| | |
|---|---|
| $t_{i,i'}$ | Traveling time between node i and i′. |
| $p_{j,s}$ | Processing time of the job j at stage s |
| $pi_j$ | Tardiness penalty in delivering job j. |
| $c_{i,i'}$ | Traveling cost between node i to i′. |
| $d_j$ | Due date of the job j. |
| $fc$ | Fixed cost of vehicle v. |
| $cap_v$ | Capacity of vehicle v. |
| $h_j$ | Holding cost of job j in unit time. |
| $M_1$ | A large number in production constraints. |
| $M_2$ | A large number. |

Variables

| | |
|---|---|
| $Z$ | Objective function |
| $co_{j,s}$ | Completion time of the job j at stage s. |
| $ar_{i,v}$ | Arriving time of vehicle v to the node i (equal to the delivery time of job $j = i - 1$ by vehicle v). |
| $r_v$ | Release time of vehicle v |
| $L_j$ | Tardiness in delivery of job j. |
| $x_{j,j',s}$ | Binary variable that is 1 when job j is produced before the job j′ at the stage s, otherwise, it is 0. |
| $y_{j,s,m}$ | Binary variable that is 1 when job j is produced on machine m at the stage s, otherwise it is 0. |

| | |
|---|---|
| $z_{i,i',v}$ | Binary variable that is 1 when the vehicle v goes from node i to node i′, otherwise, it is 0. |
| $ho_j$ | Holding time of job j. |
| $es_i$ | An auxiliary variable in the elimination sub-tour constraint. |
| $w_{i,v}$ | Binary variable that is 1 when node i is allocated to the vehicle v, otherwise equals 0. |
| $u_v$ | Binary variable that is 1 when the vehicle v is used, and otherwise it is 0. |

$$\text{Min } Z = \sum_v (fc.\,u_v) + \sum_{i,i',v} (c_{i.i'}.\,z_{i,i',v}) + \sum_j (pi_j.\,L_j) + \sum_{j,s} (hc_j.\,ho_j) \tag{1}$$

$$co_{j,s} - co_{j,s-1} \geq \sum_m (y_{j,s,m} \cdot p_{j.s}) \qquad \forall j\,, s \tag{2}$$

$$\sum_m y_{j,s,m} = 1 \qquad \forall j\,, s \tag{3}$$

$$M_1.\,(2 - y_{j,s,m} - y_{j',s,m} + x_{j,j',s}) + co_{j,s} - co_{j',s} \geq p_{js} \qquad \forall j, j', j < j', s, m \tag{4}$$

$$M_1.\,(3 - y_{j,s,m} - y_{j',s,m} + x_{j,j',s}) + co_{j',s} - co_{j,s} \geq p_{j'.s} \qquad \forall j, j', j < j', s, m \tag{5}$$

$$\sum_{i'(1<i'<N)} z_{1,i',v} = u_v \qquad \forall i, i', v \tag{6}$$

$$\sum_{i'(i'\neq i, i'>1)} z_{i,i',v} \leq u_v \qquad \forall i, 1 < i < N\,, v \tag{7}$$

$$\sum_{i(i'\neq i)} \sum_v z_{i,i',v} = 1 \qquad \forall i, 1 < i < N \tag{8}$$

$$\sum_{i(1<i<N)} z_{i,i',v} = u_v \qquad \forall i' = N, v \tag{9}$$

$$\sum_{i(i<N, i'\neq i)} z_{i,i',v} - \sum_{i(i>1, i'\neq i)} z_{i',i,v} = 0 \qquad \forall i, 1 < i' < N\,, v \tag{10}$$

$$z_{N,1,v} = 0 \qquad \forall i, i', v \tag{11}$$

$$z_{i,i',v} + z_{i',i,v} \leq 1 \qquad \forall i, i', v \tag{12}$$

$$es_i - es_{i'} + (N.\,z_{i,i',v}) \leq N - 1 \qquad \forall i, i', v \tag{13}$$

$$\sum_{i'} z_{i,i',v} \leq w_{i,v} \qquad \forall i, v \tag{14}$$

$$\sum_v w_{i,v} = 1 \qquad \forall i, 1 < i < N \tag{15}$$

$$\sum_{i\neq 0} w_{i,v} \leq cap_v \cdot u_v \qquad \forall v \tag{16}$$

$$ar_{1,v} \geq r_v \qquad \forall v, i \tag{17}$$

$$r_v \geq co_{j,k} - M_2.\,(1 - w_{i,v}) \qquad \forall i, j, v, k, i - 1 = j \tag{18}$$

$$ar_{i,v} \leq M_1.\,w_{i,v} \qquad \forall i, i > 1, v \tag{19}$$

$$L_j \geq ar_{i,v} - d_j \qquad \forall i, j, i - 1 = j, v \tag{20}$$

$$ar_{1,v} \geq (ar_{i,v} + t_{i,1}) - M_2.\,(1 - z_{i,1,v}) \qquad \forall i, i', v \tag{21}$$

$$ho_j \geq r_v - co_{j,K} - M_2.\,(1 - w_{i,v}) \qquad \forall i, j, v, j = i - 1, s \tag{22}$$

$$Z, co_{j,s}, ar_{i,v}, f_j, L_j, uu_i, ho_j, r_v \geq 0 \qquad \forall i, i', v, j, s \tag{23}$$

$$x_{j,j',s}, y_{j,s,m}, u_v, z_{i,i',v}, w_{i,v} = 0{,}1 \qquad \forall\, s, m, i, i', v, j, j' \tag{24}$$

According to the constraint (1), the objective function includes minimizing the total cost of tardiness and fixed and variable costs of transporting goods, and the cost of holding goods waiting to be sent to the customer. Constraint (2) shows that the completion time of a job in a stage, is more than its processing time in the current stage plus its total time in the previous stage. In the production scheduling section, constraint (3) indicates that each job is not processed on more than one machine in each stage. Constraints (4) and (5) are for sequencing jobs in the production phase. If the jobs j and j′ are processed on an identical machine in one stage, then, job j will be processed before job j′.

In the vehicle routing section, according to constraints (6) and (7), each node will be visited by only one vehicle (except the destination node and the origin node). Each vehicle costs a fixed amount if it has at least one allocated customer on its tour. Constraint (8) makes sure that each customer node is not visited by a vehicle more than once. Constraint (9) shows that the destination is not visited more than once by one vehicle. Considering constraint (10), for each node the number of output and input edges are equal (Except the origin and destination nodes.). Considering constraint (11), there is not any edge from the destination node to the origin node. Constraint (12) shows that each vehicle passes through an edge at most once ($z_{i,i',v}$ and $z_{i',i,v}$ do not take the value of 1 at the same time). Also, the elimination sub-tour constraint for the vehicle routing is shown in constraint (13). Constraint (14) makes sure that a vehicle will not cross any edge containing a node if that node is not assigned to that vehicle. Due to constraint (15), a customer is not allocated to more than one vehicle. Constraint (16) ensures that each vehicle is assigned orders and is filled to capacity. In this case, each node belongs to one customer and each customer has only one order. Considering constraints (17) and (18), the arrival time of each vehicle to each node is longer than the completion time of all jobs allocated to that vehicle. According to constraint (19), the arrival time is merely calculated for the customer nodes assigned to a vehicle.

Constraint (20) calculates the amount of delay in the delivery of a job, which is equal to the difference between the due date and the delivery time of the order. Constraint (21) shows that the arrival time of the vehicle to each node equals the arrival time of that vehicle to the previous node, plus the time of passing the last edge leading to that node in that vehicle's path. This constraint will be linearized.

Constraint (22) is used to calculate the holding cost of job j in stock before distribution. This cost is proportional to the difference between the completion time of the job j and the movement of the vehicle that will deliver that job. At last, constraints (23) and (24) are related to the positive and binary variables respectively.

## 4. Dominance properties

By increasing our knowledge about the problem and using the optimal solution properties for the problem, a better solution method is provided. Here are some rules to solve the problem:

**Lemma 1**. Non-delay is not necessarily the final solution to this problem.

**Proof**. Consider a non-delay solution. If we add an unnecessary amount of delay to this solution before a job is processed, the time to complete the processing of this job and all jobs after it will increase. According to Lemma 1, this may improve the solution to the problem.

**Properties 1**. In the optimal solution, a vehicle is released at the time when its last job's is complete.

**Proof**. Suppose that there is a solution in which the vehicle is released at the completion time of the last job of that vehicle. If the release time is delayed, the manufacturer must hold the finished goods in stock, which increases the holding cost and worsens the objective function value. In addition, the delivery time of customers' orders increases and may increase the tardiness penalties and thus raises the total costs.

**Properties 2**. Assume that there is a solution through which job j′ is processed on the machine m′, in stage s′. This machine will be idle for a period of T after processing the job j′, and this job is going to be delivered by vehicle v′. If $R_{v'} \geq co_{j',s'} + \sum_{s=s'+1}^{S} p_{j',s} + T'$, that $0 \leq T' \leq T$, processing of job j′ should be delayed until completed on time $co_{j',s'} + \sum_{s=s'+1}^{S} p_{j',s} + T'$.

**Proof**. Assume that $S_1$ is a solution in which machine m′ in stage s′ will be idle after processing the job j′for time T and this job is going to be delivered by vehicle v′. Assume that T′ is a period of time ($0 \leq T' \leq T$) for which we have:

$$R_{v'} \geq co_{j',s'} + \sum_{s=s'+1}^{S} p_{j,s} + T' \tag{25}$$

Assume that $S_2$ is a similar solution in which processing job j′ is delayed until completed at $co_{j',s'} + \sum_{s=s'+1}^{S} p_{j',s} + T'$. In solution $S_2$ the release time of job j′ is $R_{v'}$ too. It is clear that these two solutions are different only in the holding cost of the job j′ at the stage $s = K$. In $S_1$ this cost is equal to:

$$\text{holding cost } (S_1) = h_{j'}.(R_{v'} - co_{j',s'} - \sum_{s=s'+1}^{S} p_{j,s}) \tag{26}$$

While in $S_2$ it is equal to:

$$\text{holding cost }(S_2) = h_{j'}.\,(R_{v'} - co_{j',s'} - \textstyle\sum_{s=s'+1}^{S} p_{j,s} - T') \tag{27}$$

So $S_2$ dominates $S_1$ and the proof is completed.

**Properties 3**. Assume that in stage $s'$ there is a solution through which, jobs $j_1, j_2$ are processed on machine $m'$ sequentially and they are going to be delivered by the same vehicle. If $h_{j_2}.\,(p_{j_1,s'}) \leq h_{j_1}.\,(p_{j_2,s'})$, then in the optimal solution, machine $m'$ processes job $j_2$ immediately before job $j_1$.

**Proof**. Suppose $S_1$ is a solution in stage s ($s = s'$), the machine $m'$ processes jobs $j_1$ and $j_2$ consecutively, and processes $j_1$ before doing $j_2$ and suppose $S_2$ is a similar solution through which job$j_2$ will be processed before job $j_1$. We suppose ST is the point in time where job $j_1$ in $S_1$ and job $j_2$ in $S_2$ begin. The total cost of holding jobs $j_1$ and $j_2$ in $S_1$ is as follows:

$$\text{holding cost }(S_1) = (h_{j_1}.\,b_{j_1}) + (h_{j_2}.\,b_{j_2}) \leq h_{j_1}.\,((R_v - ST - p_{j_1,s'} - \textstyle\sum_{s=s'+1}^{S} p_{j_1,s})) + h_{j_2}.\,((R_v - ST - p_{j_1,s'} - p_{j_2,s'} - \textstyle\sum_{s=s'+1}^{S} p_{j_2,s})) \tag{28}$$

And the total holding cost of jobs $j_1, j_2$ in $S_2$ is as follows:

$$\text{holding cost }(S_2) = (h_{j_1}.\,b_{j_1}) + (h_{j_2}.\,b_{j_2}) \leq h_{j_1}.\,((R_v - ST - p_{j_1,s'} - p_{j_2,s'} - \textstyle\sum_{s=s'+1}^{S} p_{j_1,s})) + h_{j_2}.\,((R_v - ST - p_{j_2,s'} - \textstyle\sum_{s=s'+1}^{S} p_{j_2,s})) \tag{29}$$

$$\text{holding cost }(S_2) - \text{holding cost }(S_1) \leq (h_{j_1}.\,(-p_{j_2,s})) - \left(h_{j_2}.\,(-p_{j_1,s})\right) = \left(h_{j_2}.\,(p_{j_1,s})\right) - \left(h_{j_1}.\,(p_{j_2,s})\right) \leq 0 \tag{30}$$

So $S_2$ dominates $S_1$ and the proof is completed.

**Corollary 1**. In an optimal solution, a machine will not be idle after a job is processed, unless it is the last job allocated to this machine that can be processed.

## 5. Proposed hybrid algorithm PGR

Given that the integrated scheduling problem (Gupta, 1988) in NP-hard and also the vehicle routing problem is one of the NP_hard problems (Prins, 2004), the integrated problem of HFS and VRP with holding inventory is a NP_hard problem and if the dimensions of the problem are large, it is not possible to reach an acceptable solution in an acceptable time. A meta-heuristic algorithm is used to find the solution to the integrated problem in less time. So, a hybrid PSO algorithm called PGR was used to solve it in less time in this research.

To escape from the early convergence of the PSO algorithm, most of the studies on it were about diversity in searching to escape local minimums. The proposed algorithm in this paper was applied by Pan et al. (2008). Their algorithm was a discrete PSO to solve flow shop scheduling problems where the initial population was created by using the nearest neighborhood method and NEH method. Also, they used crossover and mutation operators and a new velocity equation for the particle.

In this study, to get rid of local optimums and prevent premature convergence of the PSO algorithm, new operators of the genetic algorithm, which create more diversity, were used to update the particles' positions. Due to the discrete solution space of the problem; this discrete algorithm can be suitable for solving our problem. The problem of hybrid flow shop scheduling has a vast solution space. Therefore, reducing the solution space dimension without deleting the desired solutions can be useful. So, the relevant algorithm in the initial solution representation section used EDD and ERT rules and dominance rules to search in the solution space. The suggested algorithm structure is as follows.

**Begin (Algorithm)**

**For** each particle $i \in 1 \ldots I$ do

a. Randomly initialize $x_i$
b. Apply dominance rules on $x_i$ to improve the solutions
c. Evaluate the fitness function of particle i by $f(x_i)$
d. Set $p_i = x_i$ ($p_i$ is the individual best of the particle)
e. Set $f(G) = \infty$ (G is the global best)
f. Set $t = 0$

**End** (for)

**While (**termination criteria is not met) **do**

$t = t + 1$ (Iterations)

**For** each particle $i \in 1 \ldots I$ do

g. $p_i^1$= perform random crossover on $p_i$ and $x_i$ with the $c_1$ probability

h. Evaluate the fitness function $p_i^1$
i. **If** $(f(p_i^1) < f(p_i))$
    i. $p_i \leftarrow p_i^1$
j. **End** (IF)
k. Update $p_i$
l. $p_i^2$= perform random crossover on G and $x_i$ with the $c_2$ probability
m. Evaluate the fitness function $p_i^2$
n. **If** $(f(p_i^2) < f(p_i))$
    i. $p_i \leftarrow p_i^2$
o. **End** (IF)
p. Update $p_i$
q. $p_i^3$ = perform random mutation on $x_i$ with the ω probability
r. Evaluate the fitness function $p_i^3$
s. **If** $(f(p_i^3) < f(p_i))$
    i. $p_i \leftarrow p_i^3$
t. **End** (IF)
u. Update $p_i$
v. $p_i^4$ = Apply dominance rules on $p_i$ to improve the solutions
w. Evaluate the fitness function $p_i^4$
x. $p_i \leftarrow p_i^4$
y. Update $p_i$
z. **If** $(f(p_i) < f(G))$
    i. $G \leftarrow p_i$
aa. **End** (IF)

Update G
**End** (for)
**End** (While)
**End (Algorithm)**

### 5.1. Solution representation

We have used the job-based representation which is called JBRF in Pan et al. (2013). In this method, jobs sequence is specified in the first stage. In this solution representation, the jobs' sequence is represented at the first stage with a permutation. Considering this permutation and using dispatching rules, all jobs are allocated to the machines in that stage. Each job is set to the first idle machine.
In addition, the problem of vehicle routing needs makes two decisions: determining the number of required vehicles and assigning each job to one vehicle. Alba and Dorronsoro (2004), generated a permutation of random integers for solution representation of the routing problem. The permutation created by the numbers in the range $[1, J+V-1]$ includes orders and route dividers, and in this way, customers are assigned to the routes. In this interval, J refers to the customers' number and V refers to the vehicle number. In the permutation, orders are displayed with numbers from 1 to J and route dividers with a number from $J+1$ to $J+V-1$.
In this research, a vector that includes two parts for solution representation was used. In the first part, the jobs' sequence at the first stage was determined, and in the next part, routing of vehicles was done (Moazami et al, 2021). The solution vector for five jobs and three vehicles are shown in Figure 1.
In the solution represented in Figure 1, the permutation 1-2-5-3-4 for jobs has been specified. The first part of the vector shows the job sequence in the first production stage. The jobs are allocated to the first idle machines. After completing the production process, they enter the transportation stage. Also, in another part where the solution is represented, the second and fourth jobs are transported by the first car and the first, fifth, and third jobs by another car. To divide the J number between the V vehicles, $V-1$ separator is required; therefore, to assign five customers to three vehicles, two separators are needed. In this
In this example, the numbers 6 and 7 that are greater than the customers' number refer to the separators. The numbers located between the separators' positions in the vector are the customers allocated to a vehicle.

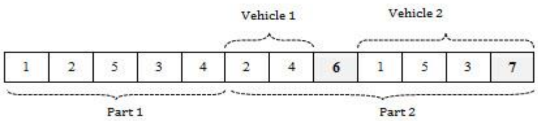

**Figure 1. The two-part vector solution representation**

### 5.2. Decoding process

In this part, the decoding of the suggested solution representation is described. In decoding the solution representation, the X vector is generated. According to the first part of the X, the sequence of jobs is determined in the first

stage, jobs are set to and scheduled on the idle machines (Matrix $Y_1$) and the completion time in the first stage is determined according to their processing time in the first stage (Matrix $P_1$). From the second stage onwards ($s > 1$), jobs sequences in the production stages are selected based on the sorted previous stage's matrix ($P_{s-1}$) and the sequence matrix for stage k is built ($Y_s$). In this way, for each solution, two matrices of size ($n_j . n_s$) are generated. The first one is Y, which shows the order in which jobs are done on the machines of each stage, and the second one is the matrix P, which represents the completion time of jobs in stages. Then the allocation of the produced jobs to the vehicles is determined considering the second part of the vector.
In Figure 2, a Gantt chart of the scheduling and vehicle routing is drawn for the solution vector presented in the previous section. The parameters $p_{j.s}$ which mean the times of production for each job performed at each stage are displayed in Table 2, and the traveling times between nodes ($t_{i,i'}$) are shown in Table 3. Machines are indicated with $M_{l.s}$, representing the machine l of the stage s. As displayed in the chart, jobs are processed on machines at the first stage according to the first part of the solution representation. After finishing their process, they go to the next stages and are assigned to the first free machines in each stage. Therefore, all jobs will be processed without any overlap. As an example, $M_{1,2}$ which means the first machine of the second stage, finishes processing jobs 2, 5, and 4 at times 6, 11, and 16, respectively.

**Table2. Processing time ($p_{j.s}$)**

| Jobs/ Stages | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 3 | 5 | 4 |
| 2 | 5 | 3 | 3 | 5 | 5 |
| 3 | 4 | 2 | 2 | 2 | 4 |

**Table3. Traveling time ($t_{i,i'}$)**

| Locations | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 4 | 3 | 4 | 2 | 5 |
| 1 | 4 | 0 | 7 | 2 | 3 | 3 |
| 2 | 3 | 7 | 0 | 7 | 4 | 2 |
| 3 | 4 | 2 | 7 | 0 | 3 | 5 |
| 4 | 2 | 3 | 4 | 3 | 0 | 2 |
| 5 | 5 | 3 | 2 | 5 | 2 | 0 |

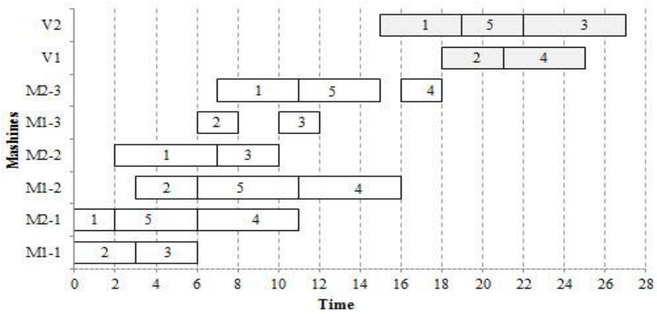


**Figure 2. Gantt chart of scheduling and vehicle routing**

So, jobs pass the stages in sequence. Finally, they are delivered to the customers (after waiting for all the assigned jobs to complete and vehicles unload). The solution's coding and decoding, guarantee to create a feasible solution that fulfills the constraints of the model such as allocation each customer to one vehicle or not visiting a customer's node more than once by vehicles and other constraints. For constraints like capacity constraint (16), penalties were defined to avoid solutions that ignore capacity limitation.

### 5.3. Initial population generation

In scheduling problems, a good initial solution has a great impact on metaheuristic algorithms. Random initial solutions are usually translated into better solutions. Therefore, the initial solution conversion method must be selected with high accuracy to achieve a high level of performance. Heuristic algorithms can be used to generate the initial population. Kaweegitbundit (2012) reviewed dispatching rules applied for HFS problems through minimizing makespan and minimizing total tardiness. He showed that to minimize the total tardiness as a goal, the rule which has better results compared to other rules is the EDD.
In this paper, the common EDD rule based on a parameter is used in the initial population. A random number is created between 0 and 1, if it is less than γ, the EDD is used to generate the initial sequence, and if it is greater than γ, the random sequence is used as the initial sequence. With the proposed decoding method, the sequence for the particle is generated. By doing so, population diversity is achieved. Then, for each particle, as specified in the solution representation section, customers are assigned to the routes. Therefore, the initial population of the algorithm is created.

### 5.4. Particles update

Since a solution includes a permutation of jobs as $(1, 2, \dots, n)$, the update equation (31) can be used for the particle (Pan et al. 2008):

Where $X_i^{\ t}$ is the position of the particle, $P_i^{\ t}$ is the best individual of that particle, and $G^t$ is the best global of the population. So, the equation consists of 4 parts:

$$X_i^{\ t} = c_2 \otimes F_3\left(c_1 \otimes F_2\left(\omega \otimes F_1\left(X_i^{\ t-1}\right), P_i^{\ t-1}\right), G^{t-1}\right) \tag{31}$$

1) The first part is $A_i^{\ t} = \omega \otimes E_1\left(X_i^{\ t-1}\right)$ which indicates the particle velocity, where $E_1$ refers to the mutation operator with the $\omega$ probability.
2) The part $B_i^{\ t} = c_1 \otimes E_2\left(A_i^{\ t}, P_i^{\ t-1}\right)$ corresponds to the individual velocity of the particle and in which $E_2$ is the crossover operator with the $c_1$ probability.
3) The part $C_i^{\ t} = c_2 \otimes E_3\left(B_i^{\ t}, G_i^{\ t-1}\right)$ corresponds to the global velocity of the particle and in which $E_3$is the crossover operator with the $c_2$ probability.

### 5.5. Mutation operator

In the proposed algorithm, a new mutation operator is defined according to the solution of the problem and with equal probability; it is applied on both parts of the presented solution vector. Two random positions in each solution part are selected in this operator. Then in the first part of the solution, the positions between these two positions are replaced by the order of those positions in the second part of the solution representation. In the second part of the solution, first, the positions between selected positions are reversed so the separators are positioned, then the positions related to the jobs between these two positions are replaced by their order in the first part of the solution. For a random selection of two positions in each part of the solution, a random number is generated between 0 and 1. If it is greater than the probability ω, the sequence of numbers between these selected positions is changed; otherwise, no change is made.

The process of this mutation operator is explained in the following example. Suppose the solution below is the mutation candidate including 7 jobs.
The selected particle is first copied exactly on the chromosome of the mutated solution. After that, two random numbers in the interval [1, J] in the first part of the solution and two random numbers in the range [1,J+V-1] are generated in the second part of the solution. Then, the sequence between these selected positions in each section is changed and transferred to the mutated chromosome in each section. In the PGR, the mutation is applied on two sections of the presented solution vector separately and simultaneously (Figure 3).

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

**Figure 3. Mutation operator**

### 5.6. Crossover operator

The main operator of the genetic algorithm is the crossover operator that reconstructs the genes of two parents and produces offspring similar to either parent. The reason for this is the transfer of good genes from one generation to the next. Therefore, it is very important to present a crossover operator that can maintain high quality part of the parents' genetic code.
In this paper, a method for crossover operator is presented according to the solution of the problem. The parent whose gene the new child's chromosome inherits is selected by using random numbers in the new crossover.
For each job, a random number is produced. When this number is less than $c_1$, the value of the first parent is copied to the child chromosome, for jobs for which the random number generated is larger than $c_1$, and the position of the jobs and separators are transferred to the child chromosome in the same order of the second parent. This crossover is first used for the flow shop and vehicle routing generation problem in the PSO algorithm, and the result of this crossover is only one child for both parents.

Stage 1: First copy the first parent to the child chromosome.

Stage 2: Do the following stages for each cell:

2-1: Generate a random number in the interval [0,1] as $C_1$.

2-2: If $C_1 > c_1$, change the order of those jobs in the first part of the child chromosome to the order in the first part of the second parent. Also, change the order of those cells in the second part of the child chromosome to the order in the second part of the second parent.

2-3: If $C_1 < c_1$, don't change the order of those cells in the child.

The following example illustrates the process for doing this crossover with a problem that has 7 jobs. (Figures 4)

The first parent chromosome is copied to the child chromosome. For each cell, a random number in the range [0,1] is generated. Suppose that the random numbers of all cells were less than $c_1$ except for the second, first, and fourth jobs in the first part of the solution and the fourth, sixth, and eighth cells in the second part of the solution. For these cells, random numbers are generated greater than $c_1$. So, the sequence of these cells must be changed. The second, first, and fourth jobs in the first part of the solution, in the second parent with the order 4-1-2, and the fourth, sixth, and eighth cells in the second part of the solution are transferred to the child in the order of 8-6-4 (Figure 5).

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parent 1 | 7 | **2** | 6 | **1** | 5 | 3 | **4** | 2 | **4** | 1 | **6** | 9 | 7 | 5 | **8** | 3 |
| Parent 2 | **4** | **1** | 7 | 3 | 6 | **2** | 5 | 3 | 7 | 5 | **8** | **6** | 9 | 1 | **4** | 2 |

**Figure 4. Crossover operator**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| offspring | 7 | **4** | 6 | **1** | 5 | 3 | **2** | 2 | **8** | 1 | **6** | 9 | 7 | 5 | **4** | 3 |

**Figure 5. Offspring in the crossover**

### 5.7. Stop condition

The proposed algorithm's stop condition is the highest number of iterations.

## 6. Computational results

The performance of the suggested algorithm is evaluated by analysis computational results. The computational results obtained from the CPLEX solver in GAMS software are compared with the algorithm solutions in MATLAB software by a computer with a 2.20 GHz processor and 6GB RAM. The comparison criterion is the mean value of the solutions for five iterations of each instance with the metaheuristic algorithm's solutions and in small problems; the solution obtained from the algorithm is compared with the CPLEX solver in GAMS software.
To generate sample problems, the problem parameters are identified. To determine the values of $n_j$, $n_m$, $n_s$ and $p_{j,s}$, the study of Oğuz et al. (2004) was useful. Ahmadizar and Farhadi's (2015) article was also used for due dates.
However, because in the case under study, if the values of that article were used, large numbers would be obtained, and this would non delay the delivery of jobs in any situation, so these values are multiplied by a factor of 0.5 to make the delivery time due dates values more compatible with other data. For the values related to routing, fc, $c_{i,i'}$ and $cap_v$, the article by Ramezanian et al. (2017) was used, and values $pi_i$ and $h_j$ were also generated randomly. These values are presented in Table 4. For each problem size, 3 instances were generated and sixty instances were created accessible at http://web.ntust.edu.tw/~ie/index.html. Customer and factory locations are also created using random latitudes and longitudes, and we can calculate the distance of points according to these locations.

### 6.1. Parameters adjusting

Parameters adjusting play a significant role in the quality of the solutions in the metaheuristic algorithm. Therefore, for the proposed algorithm, some experiments were conducted to determine the appropriate parameters for the algorithm. In the algorithm, the maximum number of iterations ($Iter_{max}$), is proportional to the production stages and the number of jobs. This parameter is equal to $10(s \times n)$. The number of populations' particles ($n_{pop}$), is set to 3n in proportion to the number of jobs. The time of each run is limited to 7200 seconds. The probability of applying the EDD and ERT rules (parameters β and γ) in determining the initial solution, respectively, are determined in the range (0.8 and 0.3). ω is the probability of the mutation operator on each particle and is in the range (0.8 and 0.2) and the probabilities $c_1$, $c_2$ are determined for applying the crossover operators on each particle and in the range (0.9 and 0.7). In order to access the best composition of these parameters, different combinations of parameters were examined. The best combination of the mentioned parameters was found by solving several instances with different problem sizes and it is presented in Table 5.

### 6.2. Comparing the results

Because no research like this research seems to exist in the literature, some PSO versions were compared with this algorithm to evaluate its performance. The comparison criterion is the average value of the five instance's solution of each problem size with the solution of other algorithms or in small problems with the GAMS solution. The comparison criteria index is called the relative deviation of the mean of the solutions from the lowest mean gained by the algorithms. This index is calculated from Equation (32).

$$RD = \frac{\text{Alg} - \text{Min}}{\text{Min}} \tag{32}$$

In this part, the suggested algorithm is compared to the metaheuristic algorithms of the literature including PSO, and Tables 6 and 7 show the results. The proposed algorithm, called PGR (PSO-GA-under dominance rules), had

five implementations for each size of the problem and the results are recorded. This algorithm uses the dominance rules and in the initial solution uses the EDD and ERT rules possibly. Particle position updates are also performed by mutation and crossover operators and dominance rules. Also, a simple version of the algorithm was tested separately for the components of the algorithm, which we call the EAPG algorithm and searches in the active solution space, also uses the crossover and mutation operators and the ERT and EDD rules randomly, but it lacks the dominance rules. Also, two other algorithms were used to analyze the effect of the algorithm's components on the results, one of them is the classical PSO and the other is the particle swarm optimization algorithm that uses just crossover and mutation operators and is called PSOGA. Since this problem is NP-hard, by increasing the dimensions of the problem, the CPLEX solver in GAMS software is not able to solve the problem in polynomial time, so, for large-size problems, the solution that GAMS reached in two hours was used in comparisons. For each size of the problem, the quality of the solution was measured by the average difference between the optimal solution and the best possible solution, and in cases where GAMS reached the optimal solution; the table is displayed with boldface in the MILP column. In these comparisons, the proposed algorithm for the problem was run with the number of iterations equal to $Iter_{max}$, and then in the period that this algorithm receives its best solution, others were executed and the results were saved to compare. For each instance of the problem, algorithms were executed five times, and then the minimum and maximum, and mean values of the objective function were presented. In small problems, the solution of the suggested algorithm was close to the exact solution; therefore, the use of this algorithm for the small size problem was logical. Due to the low error rate of the algorithm in small size instances, it can be concluded that for medium and large size problems for which no exact solution exists, the error rate is not high. Therefore, the proposed algorithm can be used to resolve large-size problems. The solutions gained by GAMS software for small sizes of the problem were compared to the result of the metaheuristic algorithm proposed in this paper and other version of the PSO algorithm in Table 6. The metaheuristic algorithms' solutions for large-scale problems are compared in Table 7.
It is clear from Table 6; PGR receives better results in less processing time in comparison to the other algorithms. The best PGR results are much better than other PSO algorithms' results, and it can be seen in the table that the PGR is superior to the others. In the Table 7, the comparison between results of the algorithms for large scale instances is shown. For proper comparison, the time is limited for each run of the algorithms equal to PGR.

**Table4. Levels of problem parameters**

| parameter | Parameter range |
|---|---|
| $n_s$: Stage numbers | 2-5-8-10 |
| : Machines numbers$n_m$ | 2-3-4-5 |
| : Job numbers$n_j$ | 5-10-20-50-100 |
| : Processing time$p_{j,s}$ | $U(1,100)$ |
| : Due date$d_j$ | $)/2((1-TAR \pm RDD/2)\sum_j \sum_k P_{j,s}$ |
| : Time of travel between two locations of the customer$t_{i,i'}$ | $x_{Depot} = y_{Depot}$ = U (20, 50)<br>$x_i$ = U (0, 2 × $x_{Depot}$ )<br>$y_i$ = U (0, 2 × $y_{Depot}$ )<br>$dis_{i,i'} = \sqrt{(x_{i'}-x_i)^2+(y_{i'}-y_i)^2} = t_{i,i'}$ |
| : Travel cost between two customer's location$c_{i,i'}$ | Unitcost = $U(50,200)$<br>= Unitcost $*$ $dis_{i,i'}c_{i,i'}$ |
| : Vehicle's capacity$cap_v$ | Small-scale 4<br>Medium-scale 10<br>Large-scale 15 |
| : Vehicle's fix cost$fc$ | $U(150,210)$ |
| : Holding cost$hc_j$ | $U(10,15)$ |
| : Tardiness penalties$pi_i$ | $U(5,15)$ |

**Table5. Parameters Adjustment**

| Value | Parameter | Notation |
|---|---|---|
| 0.5 | Possibility of the EDD | $\gamma$ |
| 0.2 | Possibility of the ERT | $\beta$ |
| 0.5 | Possibility of the mutation | ω |
| 0.8 | Possibility of the crossover with the best personal solution | $c_1$ |
| 0.9 | Possibility of the crossover with the best global solution | $c_2$ |

According to Tables 6 and 7, when the problem size increases, the presented model becomes difficult to use and large instances cannot be dealt with optimally within a logical computation time. PGR and EAPG algorithms performed better than PSOGA and PSO algorithms. Finally, from the numerical results, it is obvious that the PGR, which uses the dominance rules, has the best performance. Compared to EAPGs and PGRs, their results were roughly the same for size examples, but by increasing the jobs number to more than twenty, in some cases, the differences between PGR and EAPG results increased. These differences confirmed the significant role of the dominance rules in finding the desired solution. Also, from Tables 6 and 7, PGR is superior to PSO in 20 problems, superior to PSOGA in 18 problems, and superior to EAPG in 15 problems out of 21 problem sizes. On average, PGR performs better than the other three algorithms (with an average of 0.883).

**Table6. Solutions in small size problems**

| Problem n-m-s | Instance | MILP Qu | MILP Ti | PSO Mi | PSO Av | PSO Ma | PSOGA Mi | PSOGA Av | PSOGA Ma | EAPG Mi | EAPG Av | EAPG Ma | PGR Mi | PGR Av | PGR Ma | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3-2-2 | 1 | **0** | 0.21 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 2 |
| | 2 | **0** | 0.31 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 1 |
| | 3 | **0** | 0.14 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 1 |
| | Ave | 0 | 0.33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 5-2-2 | 1 | **0** | 209 | **0** | 0.02 | 0.08 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 2 |
| | 2 | **0** | 191 | **0** | 0.01 | 0.07 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 1 |
| | 3 | **0** | 119 | 0.04 | 0.07 | 0.12 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 3 |
| | Ave | 0 | 204 | 0.01 | 0.03 | 0.09 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 5-2-5 | 1 | **0** | 1183 | 0.19 | 0.29 | 0.47 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 4 |
| | 2 | **0** | 950 | 0.25 | 0.46 | 0.62 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 3 |
| | 3 | **0** | 1023 | 0.13 | 0.18 | 0.21 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 3 |
| | Ave | 0 | 1052 | 0.19 | 0.31 | 0.43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 5-2-8 | 1 | **0** | 7199 | **0** | 0.2 | 0.05 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 3 |
| | 2 | **0** | 7200 | 0.28 | 0.31 | 0.58 | **0** | 0.01 | 0.04 | **0** | **0** | **0** | **0** | **0** | **0** | 2 |
| | 3 | **0** | 7199 | 0.08 | 0.15 | 0.26 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 4 |
| | Ave | 0 | 7199 | 0.12 | 0.22 | 0.30 | 0 | 0 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 5-2-10 | 1 | 0.15 | 36 | 0.37 | 1.08 | 2.02 | 0 | 0.01 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 | 6 |
| | 2 | **0** | 283 | 0.09 | 0.29 | 0.46 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | 5 |
| | 3 | **0** | 3015 | 0.08 | 0.26 | 0.37 | 0.01 | 0.03 | 0.06 | **0** | **0** | **0** | **0** | **0** | **0** | 7 |
| | Ave | 0.05 | 1111 | 0.18 | 0.54 | 0.95 | 0 | 0.01 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 6 |
| 10-3-2 | 1 | 0.21 | 4 | 0.15 | 0.17 | 0.21 | 0 | 0.06 | 0.26 | 0 | 0 | 0 | 0 | 0 | 0 | 15 |
| | 2 | 0.04 | 3 | 0.16 | 0.24 | 0.32 | 0 | 0.03 | 0.09 | 0 | 0 | 0 | 0 | 0 | 0 | 18 |
| | 3 | 0.16 | 5 | 0.18 | 0.24 | 0.31 | 0 | 0 | 0.03 | 0 | 0.01 | 0.05 | 0 | 0 | 0 | 17 |
| | Ave | 0.14 | 4 | 0.16 | 0.22 | 0.28 | 0 | 0.03 | 0.13 | 0 | 0 | 0.02 | 0 | 0 | 0 | 16.7 |
| 10-3-5 | 1 | 0.05 | 3 | 0.05 | 0.10 | 0.16 | 0 | 0 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 | 31 |
| | 2 | 0.31 | 6 | 0.58 | 1.5 | 1.93 | 0.09 | 0.13 | 0.19 | 0.01 | 0.05 | 0.09 | 0 | 0 | 0 | 33 |
| | 3 | 0.13 | 5 | 0.21 | 0.23 | 0.29 | 0.06 | 0.07 | 0.09 | 0 | 0 | 0 | 0 | 0 | 0 | 35 |
| | Ave | 0.16 | 4.67 | 0.28 | 0.61 | 0.80 | 0.05 | 0.07 | 0.1 | **0** | 0.02 | 0.03 | 0 | 0 | 0 | 33 |
| 10-3-8 | 1 | 0.81 | 7 | 0.93 | 2.09 | 3.25 | 0.11 | 0.23 | 0.32 | 0.01 | 0.18 | 0.22 | 0 | 0 | 0 | 52 |
| | 2 | 0.74 | 9 | 0.52 | 1.58 | 3.34 | 0 | 0.05 | 0.08 | 0 | 0 | 0 | 0 | 0 | 0 | 48 |
| | 3 | 0.12 | 13 | 0.43 | 0.58 | 0.69 | 0 | 0.03 | 0.09 | 0 | 0.03 | 0.09 | 0 | 0 | 0 | 55 |
| | Ave | 0.57 | 9.7 | 0.63 | 1.42 | 2.43 | 0.04 | 0.10 | 0.16 | 0 | 0.07 | 0.10 | 0 | 0 | 0 | 52 |
| 10-3-10 | 1 | 0 | 14 | 0.52 | 0.67 | 0.89 | 0.04 | 0.09 | 0.15 | 0.02 | 0.05 | 0.08 | 0.01 | 0.02 | 0.04 | 106 |
| | 2 | 1.35 | 25 | 0.34 | 0.59 | 1.08 | 0.08 | 0.11 | 0.15 | 0 | 0.06 | 0.14 | 0 | 0 | 0 | 44 |
| | 3 | 1.25 | 15 | 1.56 | 1.86 | 2.41 | **0** | 0.01 | 0.03 | **0** | **0** | **0** | **0** | 0.01 | 0.03 | 43 |
| | Ave | 0.87 | 18 | 0.81 | 1.04 | 1.46 | 0.04 | 0.07 | 0.11 | 0.001 | 0.04 | 0.07 | 0.003 | 0.01 | 0.02 | 43 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Average | | 20.0 | 1066 | 0.26 | 0.49 | 0.75 | 0.01 | 0.03 | 0.07 | 0 | 0.01 | 0.02 | 0 | 0.001 | 0.002 | 17.74 |

**Table 7. Solutions in medium and large size problem**

| Problem n-m-s | Instance | MILP | | PSO | | | PSOGA | | | EAPG | | | PGR | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Quality | Time | Min | Ave | Max | Min | Ave | Max | Min | Ave | Max | Min | Ave | Max | Time |
| 20-4-2 | 1 | 0 | 125 | 0.97 | 1.25 | 1.36 | 0.11 | 0.14 | 0.57 | 0.16 | 0.19 | 0.24 | 0.02 | 0.03 | 0.05 | 49 |
| | 2 | 0 | 112 | 0.87 | 0.91 | 0.98 | 0.36 | 0.43 | 0.51 | 0.06 | 0.09 | 0.14 | 0 | 0.01 | 0.02 | 41 |
| | 3 | 0 | 179 | 0.73 | 0.79 | 0.84 | 0.19 | 0.24 | 0.26 | 0.01 | 0.14 | 0.18 | 0.01 | 0.02 | 0.04 | 52 |
| | Ave | 0 | 139 | 0.86 | 0.93 | 1.06 | 0.22 | 0.27 | 0.45 | 0.08 | 0.14 | 0.18 | 0.01 | 0.02 | 0.04 | 47 |
| 20-4-5 | 1 | 1.62 | 45 | 1.01 | 1.19 | 1.51 | 0.22 | 0.27 | 0.34 | 0.06 | 0.14 | 0.21 | 0 | 0.01 | 0.03 | 116 |
| | 2 | 0.55 | 173 | 0.28 | 0.46 | 0.58 | 0.84 | 1.06 | 1.68 | 0 | 0.05 | 0.11 | 0.02 | 0.04 | 0.08 | 121 |
| | 3 | 2.41 | 198 | 0.55 | 0.67 | 0.79 | 0.32 | 0.39 | 0.44 | 0.17 | 0.19 | 0.23 | 0.01 | 0.04 | 0.07 | 115 |
| | Ave | 1.53 | 262 | 0.61 | 0.77 | 0.96 | 0.46 | 0.57 | 0.82 | 0.08 | 0.13 | 0.18 | 0.01 | 0.03 | 0.06 | 117 |
| 20-4-8 | 1 | 0.71 | 638 | 0.45 | 0.62 | 0.83 | 0.75 | 0.84 | 0.88 | 0.73 | 0.79 | 0.89 | 0 | 0.04 | 0.07 | 229 |
| | 2 | 0.92 | 446 | 0.47 | 0.57 | 0.65 | 0.52 | 0.58 | 0.65 | 0.39 | 0.46 | 0.66 | 0 | 0.02 | 0.04 | 253 |
| | 3 | 0 | 718 | 0.53 | 0.59 | 0.68 | 0.23 | 0.27 | 0.29 | 0.24 | 0.26 | 0.28 | 0.02 | 0.09 | 0.15 | 265 |
| | Ave | 0.54 | 601 | 0.48 | 0.59 | 0.72 | 0.50 | 0.56 | 0.61 | 0.45 | 0.50 | 0.61 | 0 | 0.05 | 0.09 | 249 |
| 20-4-10 | 1 | 1.05 | 1240 | 0.82 | 0.93 | 1.16 | 0.63 | 0.77 | 0.87 | 0.35 | 0.39 | 0.45 | 0 | 0.10 | 0.13 | 943 |
| | 2 | 1.31 | 1740 | 0.61 | 0.75 | 1.03 | 0.49 | 0.59 | 0.68 | 0.23 | 0.31 | 0.39 | 0 | 0.12 | 0.16 | 862 |
| | 3 | 0.86 | 450 | 0.60 | 0.75 | 0.96 | 0.32 | 0.41 | 0.53 | 0.26 | 0.28 | 0.34 | 0 | 0.03 | 0.09 | 964 |
| | Ave | 1.07 | 1143 | 0.68 | 0.81 | 1.05 | 0.48 | 0.59 | 0.70 | 0.28 | 0.33 | 0.39 | 0 | 0.08 | 0.13 | 923 |
| 50-5-2 | 1 | 1.21 | 3200 | 0.45 | 0.53 | 0.68 | 0.11 | 0.15 | 0.27 | 0.02 | 0.09 | 0.12 | 0 | 0.02 | 0.06 | 434 |
| | 2 | 1.12 | 7200 | 0.63 | 0.75 | 0.88 | 0.20 | 0.35 | 0.44 | 0.03 | 0.04 | 0.07 | 0 | 0.01 | 0.07 | 495 |
| | 3 | 1.58 | 4235 | 0.72 | 0.86 | 0.91 | 0.51 | 0.57 | 0.59 | 0.07 | 0.08 | 0.11 | 0 | 0.02 | 0.04 | 388 |
| | Ave | 1.30 | 4878 | 0.60 | 0.71 | 0.82 | 0.27 | 0.36 | 0.43 | 0.04 | 0.07 | 0.10 | 0 | 0.02 | 0.06 | 325 |
| 50-5-5 | 1 | 1.43 | 7200 | 0.85 | 1.07 | 1.43 | 0.07 | 0.13 | 0.18 | 0.05 | 0.09 | 0.13 | 0 | 0.03 | 0.08 | 1221 |
| | 2 | 0.58 | 5247 | 0.52 | 1.41 | 2.44 | 0.09 | 0.19 | 0.25 | 0.15 | 0.21 | 0.35 | 0 | 0.05 | 0.03 | 1129 |
| | 3 | 1.42 | 4200 | 1.18 | 2.02 | 3.15 | 0.17 | 0.24 | 0.31 | 0.11 | 0.27 | 0.37 | 0 | 0.04 | 0.16 | 1260 |
| | Ave | 1.11 | 4878 | 0.85 | 1.50 | 2.34 | 0.11 | 0.19 | 0.25 | 0.10 | 0.19 | 0.28 | 0 | 0.04 | 0.09 | 1203 |
| 50-5-8 | 1 | 0.68 | 7200 | 0.73 | 0.82 | 0.95 | 0.26 | 0.29 | 0.33 | 0. | 0.22 | 0.29 | 0.03 | 0.05 | 0.07 | 2821 |
| | 2 | 1.52 | 7200 | 1.21 | 1.47 | 2.41 | 0.33 | 0.44 | 0.59 | 0.08 | 0.13 | 0.19 | 0 | 0.01 | 0.06 | 2129 |
| | 3 | 1.12 | 7200 | 0.95 | 1.08 | 1.26 | 0.28 | 0.30 | 0.34 | 0.10 | 0.18 | 0.26 | 0 | 0.01 | 0.04 | 2200 |
| | Ave | 2.41 | 7200 | 0.96 | 1.12 | 1.54 | 0.29 | 0.34 | 0.42 | 0.11 | 0.18 | 0.25 | 0.01 | 0.02 | 0.06 | 2383 |
| 50-5-10 | 1 | 1.27 | 7200 | 0.83 | 0.95 | 0.99 | 0.25 | 0.29 | 0.35 | 0.19 | 0.22 | 0.29 | 0 | 0.05 | 0.12 | 3405 |
| | 2 | 1.51 | 7200 | 1.15 | 1.21 | 1.49 | 0.09 | 0.13 | 0.21 | 0.12 | 0.16 | 0.22 | 0 | 0.02 | 0.07 | 3579 |
| | 3 | 1.69 | 7200 | 0.98 | 1.16 | 1.42 | 0.19 | 0.26 | 0.33 | 0.19 | 0.36 | 0.55 | 0 | 0.02 | 0.06 | 3509 |
| | Ave | 1.49 | 7200 | 0.99 | 1.11 | 1.30 | 0.18 | 0.23 | 0.30 | 0.17 | 0.24 | 0.35 | 0 | 0.03 | 0.08 | 3497 |
| 100-5-2 | 1 | 1.52 | 7200 | 0.89 | 0.94 | 1.12 | 0.42 | 0.56 | 0.64 | 0.16 | 0.22 | 0.29 | 0 | 0.05 | 0.08 | 3754 |

| Problem n-m-s | Instance | MILP | | PSO | | | PSOGA | | | EAPG | | | PGR | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Quality | Time | Min | Ave | Max | Min | Ave | Max | Min | Ave | Max | Min | Ave | Max | Time |
| | 2 | 2.26 | 7200 | 1.04 | 1.13 | 1.25 | 0.39 | 0.46 | 0.51 | 0.18 | 0.20 | 0.24 | 0 | 0.04 | 0.07 | 3579 |
| | 3 | 1.31 | 7200 | 0.87 | 0.91 | 0.98 | 0.55 | 0.59 | 0.65 | 0.18 | 0.22 | 0.28 | 0 | 0.02 | 0.06 | 3662 |
| | Ave | 1.70 | 7200 | 0.93 | 0.99 | 1.12 | 0.45 | 0.54 | 0.60 | 0.17 | 0.21 | 0.27 | 0 | 0.04 | 0.07 | 3665 |
| 100-5-5 | 1 | 2.79 | 7200 | 0.97 | 1.15 | 1.25 | 0.33 | 0.36 | 0.38 | 0.14 | 0.16 | 0.19 | 0 | 0.08 | 0.09 | 5268 |
| | 2 | 2.51 | 7200 | 0.96 | 1.05 | 1.09 | 0.64 | 0.75 | 0.85 | 0.16 | 0.24 | 0.29 | 0 | 0.04 | 0.07 | 5075 |
| | 3 | 1.60 | 7200 | 1.13 | 1.15 | 1.22 | 0.32 | 0.36 | 0.39 | 0.09 | 0.15 | 0.20 | 0 | 0.03 | 0.08 | 5561 |
| | Ave | 2.30 | 7200 | 1.02 | 1.12 | 1.19 | 0.43 | 0.49 | 0.54 | 0.13 | 0.18 | 0.23 | 0 | 0.05 | 0.08 | 5301 |
| 100-5-8 | 1 | 2.02 | 7200 | 0.99 | 1.21 | 1.33 | 0.26 | 0.29 | 0.35 | 0.17 | 0.19 | 0.22 | 0 | 0.05 | 0.08 | 7200 |
| | 2 | 1.87 | 7200 | 1.15 | 1.25 | 1.36 | 0.54 | 0.62 | 0.74 | 0.18 | 0.29 | 0.32 | 0 | 0.04 | 0.08 | 7200 |
| | 3 | 2.13 | 7200 | 1.05 | 1.38 | 1.96 | 0.27 | 0.30 | 0.35 | 0.09 | 0.16 | 0.20 | 0 | 0.05 | 0.07 | 7200 |
| | Ave | 2.00 | 7200 | 1.06 | 1.28 | 1.55 | 0.36 | 0.40 | 0.48 | 0.15 | 0.21 | 0.25 | 0 | 0.05 | 0.08 | 7200 |
| 100-5-10 | 1 | 2.63 | 7200 | 1.63 | 1.69 | 1.75 | 0.44 | 0.48 | 0.55 | 0.08 | 0.09 | 0.16 | 0 | 0.04 | 0.08 | 7200 |
| | 2 | 2.61 | 7200 | 1.65 | 1.71 | 1.77 | 0.43 | 0.48 | 0.53 | 0.04 | 0.06 | 0.08 | 0 | 0.04 | 0.06 | 7200 |
| | 3 | 2.15 | 7200 | 1.71 | 1.76 | 1.82 | 0.48 | 0.53 | 0.56 | 0.05 | 0.07 | 0.11 | 0 | 0.05 | 0.07 | 7200 |
| | Ave | 2.63 | 7200 | 1.66 | 1.72 | 1.78 | 0.45 | 0.50 | 0.55 | 0.06 | 0.07 | 0.12 | 0 | 0.04 | 0.07 | 7200 |
| Average | | 1.39 | 4238 | 0.82 | 0.97 | 1.19 | 0.32 | 0.39 | 0.47 | 0.14 | 0.19 | 0.25 | 0.002 | 0.04 | 0.07 | 2470 |

Also, in order to evaluate the differences between the algorithms' results, some two-tail paired t-tests were performed with SPSS software. The tests' results and the comparison based on the deviation from the best solution for the algorithms are presented in Table 8. These tests were performed at a confidence level of 95%. Zero hypotheses ($H_0$) means the absence of difference between algorithms and alternative hypotheses ($H_1$) shows the difference between algorithms. Since the value, Sig ($p-$ value) for tests is less than the type one error ($\alpha$) which is equal to $0.05$ and both upper and lower bounds ranges do not include 0, the null hypothesizes are rejected for all of these tests. According to the results of t-tests, PGR is superior to the other three algorithms with a strong statistical significance level.

**Table 8. The statistical significance level of the difference in large size instances**

| Pair | | Average | Standard Deviation | 95% Confidence Interval of the Difference | | Sig. (2-tailed) |
|---|---|---|---|---|---|---|
| | | | | Lower | Upper | |
| 1 | PSO - PGR | 0.93167 | 0.42544 | 0.66136 | 1.20198 | 0.000011 |
| 2 | PSOGA - PGR | 0.38083 | 0.13215 | 0.29687 | 0.46480 | 0.00008 |
| 3 | EAPG - PGR | 0.16500 | 0.10859 | 0.09601 | 0.23399 | 0.00027 |

### 6.3. Managerial insights

In order to obtain a proper managerial view of the problem, a number of examples of the problem in different dimensions have been examined. The parameters under consideration are holding cost, vehicle capacity and fixed cost of vehicles. An appropriate perspective can be appropriate when transportation and production planning, for example when choosing the size and number of vehicles.

To show the effect of changing the capacity and fixed cost of vehicles and holding costs on the rate of vehicles usage, the rates of these parameters were changed to 10, 30 and 50% of the initial value and several problems of medium and large dimensions were solved, the results can be seen in Figure 6. As the capacity of vehicles increases, the number of jobs carried out by each vehicle increases, so the percentage of vehicle usage increases. Therefore, the number of vehicles required may be reduced and fixed transportation costs may be saved. The more instance customers there are, the higher the usage percentage increase. The reason for this is that in the case of low number of customers, the capacity of vehicles is higher than the total amount of orders, and increasing the capacity of vehicles does not have much effect on the percentage of vehicle usage.

The second parameter to consider is the fixed transportation cost. When the cost of transportation variable is relatively small, the number of jobs transported by vehicles and consequently the percentage of these vehicles' usage may decrease. Therefore, the number of vehicles required may increase but there will be savings in tardiness costs. The more instance customers there are, the lower the usage percentage reduction. When the number of customers is high, the percentage of vehicle usage is generally high. Thus, the effect of fixed cost reduction becomes more apparent in the case of issues with fewer customers.

The third parameter studied was the holding cost rate. When the holding cost rate is relatively small, most fixed and variable transportation costs affect how work is delivered. In addition, the number of jobs transported by vehicles can be increased and thus the percentage of use of these vehicles. Therefore, the number of vehicles required may be reduced and transportation costs will be saved. The fewer instance customers there are, the higher the usage percentage increase. The reason for this is that in the case of problems with a large number of customers, the percentage of vehicle usage is generally high, so the chances of increasing it again are less because the capacity of vehicles is limited.

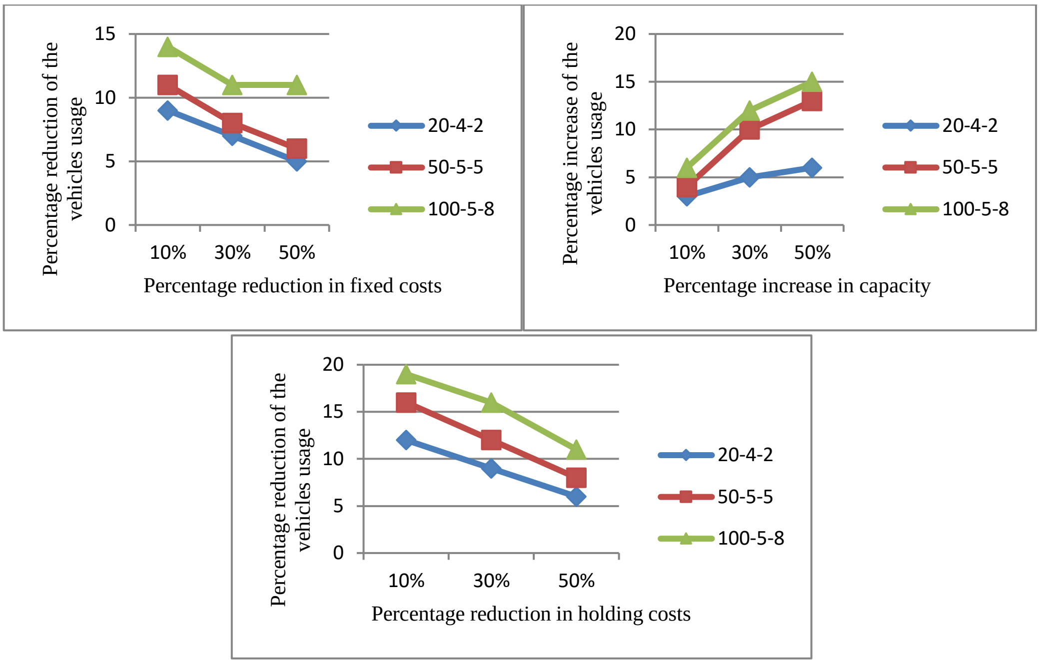


**Figure 6. Percentage change of the vehicles usage by changing the parameters related to vehicles**

## 7. Conclusions

In this paper, a new model has been proposed in which the holding cost is integrated into the hybrid flow shop production system scheduling and the vehicle routing to minimize costs of transportation and holding inventory and tardiness penalties. In a series of production environments, there are a number of same parallel machines at each stage. In the distribution sector, there are several limited capacity vehicles. Given that the integrated problem of HFS and VRP with holding inventory is an NP_hard problem, it is not possible to reach an acceptable solution in an acceptable time for large size instances. A meta-heuristic algorithm is used to find the solution to the integrated problem in less time. So, a hybrid PSO algorithm called PGR that utilizes new crossover and mutation operators and dominance rules were put forward to solve the integrated problem in less time in this research. Due to the novelty of this subject literature, randomly generated problems were used to measure the efficiency and effectiveness of the proposed solution method. To show the efficiency of the presented solution method, metaheuristics presented in the literature were used includes PSO, PSOGA and EAPG. The time of each run of the algorithms is limited to 7200 seconds. Since the relative deviation of the mean of the solutions from the lowest mean is the main factor in this study, the algorithm with lower RD is more efficient. The results of computational comparison showed that the suggested solution method provides relatively good results compared to the other algorithms in the literature. The PGR algorithm can find well-quality solutions to the investigated sample problems with the average of relative deviation 0.004 in compare with other algorithms with average of relative deviation 0.19, 0.39, 0.97 and 1.39 in the same time in large size instances.

As a suggestion for future research, we can also notice the integrated scheduling of hybrid flow shop and vehicle routing in other conditions of the production, such as different machines with different numbers in the stages of the hybrid flow shop production environment. Also, we could refer to the problem of semi-finished goods' inventory during production and between the production stages and their holding costs.

## References

Aghezzaf, E. H., & Van Landeghem, H. (2002). An integrated model for inventory and production planning in a two-stage hybrid production system. International journal of production research, 40(17), 4323-4339.

Agnetis, A., Aloulou, M. A., & Kovalyov, M. Y. (2017). Integrated production scheduling and batch delivery with fixed departure times and inventory holding costs. International Journal of Production Research, 55(20), 6193-6206.

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, April). Solving the vehicle routing problem by using cellular genetic algorithms. In European Conference on Evolutionary Computation in Combinatorial Optimization (pp. 11-20). Springer, Berlin, Heidelberg.

Ayough A., Khorshidvand B., (2019), "Designing a Manufacturing Cell System by Assigning Workforce". Journal of Industrial Engineering and Management. Vol. 12, No. 1, pp. 13-26.

Fu, Y., Tian, G., Fathollahi-Fard, A. M., Ahmadi, A., & Zhang, C. (2019). Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint. Journal of cleaner production, 226, 515-525.

Gupta, J. N. (1988). Two-stage, hybrid flowshop scheduling problem. Journal of the operational Research Society, 39(4), 359-364.

Hao, J. H., Li, J. Q., Du, Y., Song, M. X., Duan, P., & Zhang, Y. Y. (2019). Solving distributed hybrid flowshop scheduling problems by a hybrid brain storm optimization algorithm. Ieee Access, 7, 66879-66894.

Izadi, L., Ahmadizar, F., & Arkat, J. (2018). A hybrid imperialist competitive algorithm for integrated scheduling of production and distribution with vehicle routing. Journal of Industrial Engineering Research in Production Systems, 6(12), 63-81.

Izadi, L., Ahmadizar, F., & Arkat, J. (2020). A Hybrid Genetic Algorithm for Integrated Production and Distribution Scheduling Problem with Outsourcing Allowed. International Journal of Engineering, 33(11), 2285-2298.

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(1), 238-245.

Karimi, N., & Davoudpour, H. (2017). Integrated production and delivery scheduling for multi-factory supply chain with stage-dependent inventory holding cost. Computational and Applied Mathematics, 36(4), 1529-1544.

Kaweegitbundit, P. (2012). Evaluation dispatching rules for two-stage hybrid flow shop scheduling with parallel machines. In Applied Mechanics and Materials (Vol. 152, pp. 1487-1491). Trans Tech Publications Ltd.

Meinecke, C., & Scholz-Reiter, B. (2014). A heuristic for the integrated production and distribution scheduling problem. Int. Sci. Index, 8(2), 290-297.

Moazami R., Ahmadizar F., farughi H., (2021), "Integrated hybrid flow shop scheduling and vehicle routing problem", Journal of Industrial and Systems Engineering, Vol. 13, No. 2, pp. 223-244.

Mohammadi, S., Al-e-Hashem, S. M., & Rekik, Y. (2020). An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company. International Journal of Production Economics, 219, 347-359.

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.

Mousavi, M., Hajiaghaei–Keshteli, M., & Tavakkoli–Moghaddam, R. (2020). Two calibrated meta-heuristics to solve an integrated scheduling problem of production and air transportation with the interval due date. Soft Computing, 24(21), 16383-16411.

Oğuz, C., Zinder, Y., 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. C. (2008). A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. Computers & Operations Research, 35(9), 2807-2839.

Pan, Q. K., Wang, L., Mao, K., Zhao, J. H., & Zhang, M. (2012). An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process. IEEE Transactions on Automation Science and Engineering, 10(2), 307-322.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & operations research, 31(12), 1985-2002.

Ramezanian, R., Mohammadi, S., & 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, (2015), Seville, Spain.

Scholz-Reiter, B., Makuschewitz, T., Novaes, A. G. N., Frazzon, E. M., & Lima Junior, O. F., (2011), "An approach for the sustainable integration of production and transportation scheduling". International Journal of Logistics Systems and Management, Vol. 10, No. 2, pp. 158-179. doi:10.1504/IJLSM.2011.042626.

Shabtay, D. (2010). Scheduling and due date assignment to minimize earliness, tardiness, holding, due date assignment and batch delivery costs. International Journal of Production Economics, 123(1), 235-242.

Taghizadehalvandi, M., & Kamisli Ozturk, Z. (2019). Multi-objective solution approaches for employee shift scheduling problems in service sectors (research note). International Journal of Engineering, 32(9), 1312-1319.

Ullrich, C. A. (2013). Integrated machine scheduling and vehicle routing with time windows. European Journal of Operational Research, 227(1), 152-165.

Wang, D. Y., Grunder, O., & Moudni, A. E. (2015). Integrated scheduling of production and distribution operations: a review. International Journal of Industrial and Systems Engineering, 19(1), 94-122.

**This article can be cited:** Ahmadizar, F., Moazemi Goodarzi, R., Farughi,F., (2022). Hybrid flow shop scheduling and vehicle routing by considering holding costs. Journal of Industrial Engineering and Management Studies, Vol.9, No. 1, pp. 63-80.