

# Uncertain Supply Chain Management

homepage: [www.GrowingScience.com/uscm](http://www.GrowingScience.com/uscm)

## Coordinating order acceptance and integrated lot streaming-batch delivery scheduling considering third party logistics

Amir Noroozi<sup>a</sup>, Mohammad Mahdavi Mazdeh<sup>a\*</sup>, Mehdi Heydari<sup>a</sup> and Morteza Rasti-Barzoki<sup>b</sup>

<sup>a</sup>Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

<sup>b</sup>Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

### CHRONICLE

#### Article history:

Received March 18, 2018

Accepted May 8 2018

Available online

May 8 2018

#### Keywords:

Genetic algorithm

Order acceptance

Flexible flow shop

Lot streaming

Batch delivery

Third-party logistics

### ABSTRACT

Inspired by the industries such as food and beverage, metal and steel, as well as petroleum and petrochemical ones, the current study addresses a joint order acceptance and scheduling, lot streaming in a flexible flow shop and batch delivery problem. For maximizing a profit objective function with trading off between the revenue of the accepted orders and the costs incurred, a novel mixed integer linear programming is proposed. This paper develops a hybrid metaheuristic algorithm based on the Genetic Algorithm. In the developed algorithm, (1) a heuristic, (2) a local search, and (3) a restart phase is proposed. To set the appropriate parameters of the algorithms, Taguchi experimental design was applied. The obtained results reveal the appropriate performance of the hybrid genetic algorithm.

© 2019 by the authors; licensee Growing Science, Canada

## 1. Introduction

In the real world, the cost of accepting an order may be greater than its revenue and therefore, a company should not accept all orders received. The production and distribution capacity may not be sufficient for production and delivery of all the orders and it may lead to loss of its credibility. This problem has been introduced in the literature as Order Acceptance (OA). In the order acceptance problem, the capacity has a key role. Besides, an effective scheduling can improve the capacity. So, the scheduling of the accepted orders is important to achieve business goals: Order Acceptance and Scheduling (OA&S). Slotnick (2011) has presented a review of OA&S literature. Silva et al. (2018) considered the OA&S with sequence-dependent setup times on a single machine. Chaurasia and Singh (2017) considered the same problem by considering release dates and sequence dependent setup times. This problem, with customer class set up, was investigated by Xie and Wang (2016). The OA&S was addressed in a two machine flow shops by Esmailbeigi et al. (2016). Thevenin et al. (2016) studied the OA&S in a single machine for minimizing the earliness and tardiness. Emami et al. (2016) studied OA&S in non-identical parallel machines environment in which the processing times are uncertain.

\* Corresponding author Tel.: +98 21 73225002

E-mail address: [mazdeh@iust.ac.ir](mailto:mazdeh@iust.ac.ir) (M. Mahdavi Mazdeh)

All of the above review papers considered OA&S for the single machine, parallel machine, and flow shop, while, in highly realistic scheduling environments such as food and beverage, steel and metal as well as petroleum and petrochemical ones, there are multi-stages with multi-processors, meaning flexible flow shop (Ahonen & de Alvarenga, 2017; Tang et al., 2016).

In the flexible flow shop environment, in addition to the scheduling of jobs, management of production flow or lot streaming (LS) is important. In the LS, a lot of the orders can be splits into smaller sub-lots for processing that can lead to increasing the machine productivity and accelerating the production (Cheng et al., 2013). Božek and Werner (2017) studied the flexible job shop LS problem to minimize the makespan. Zhang et al. (2017) addressed the LS in a hybrid flow shop to minimize the flow time. Lalitha et al. (2017) and Ming Cheng et al. (2016) considered the LS in the same environment to minimize the makespan. Mukherjee et al. (2017) investigated the LS in a two machine flow shop by considering sub-lot-attached setup time. The objective was to determine number of sub-lots and sub-lot sizes and minimize makespan. Table 1 summarizes the assumptions and features of some of recently closely related papers of OA&S and LS.

**Table 1**

Summary of the closely related OA&S and LS

Ref.	OA&S	LS	Production Environment*				Objective**			Solution Approach***		
			S	P	FS	FFS	Service	Cost	Revenue	DP	B&B	Meta
(Reisi-Nafchi & Moslehi, 2015a,b)	✓		✓				✓		✓			GA
(Reisi-Nafchi & Moslehi, 2015)	✓		✓				✓		✓	✓		
(Emami et al., 2016)	✓			✓			✓		✓			Lagrangian relaxation
(Chaurasia & Singh, 2016)	✓		✓				✓		✓			GA
(Ou & Zhong, 2016)	✓			✓			✓	✓				Heuristic
(Nguyen, 2016)	✓		✓				✓		✓			GA
(Lei & Guo, 2015)	✓				✓		✓		✓			PNS
(C. Chen et al., 2014)	✓		✓				✓		✓			GA
(Wang et al., 2013)	✓				✓		✓		✓		✓	
(Cesaret et al., 2012)	✓				✓		✓		✓			TS
(Noroozi et al., 2017)	✓		✓				✓	✓	✓			GA+PSO
(Lalitha et al., 2017)		✓				✓	✓					Heuristic
(Zhang et al., 2017)		✓				✓						MBO
(Han et al., 2016)		✓			✓							NSGA-II
(Mukherjee et al., 2017)		✓			✓		✓					Optimal properties
(Nejati et al., 2016)		✓				✓						GA&SA
(Ming Cheng et al., 2016)		✓				✓						Heuristic
(Sang et al., 2015)		✓			✓							IWO

\* S: Single machine ; P: Parallel machine; F: Flow Shop; FFS: Flexible Flow Shop

\*\* Service: the time-based, Cost: cost-based, and Revenue: revenue-based functions

\*\*\*SA: Simulated Annealing ; PSO: Particle Swarm Optimization ; TS: Tabu search ; MBO: Migrating Birds optimization ; NSGA-II: Non-dominated Sorting Genetic Algorithm II ; IWO: Invasive Weed Optimization

DP: Dynamic Programming; B&B: Branch & Bound; Meta: Metaheuristic

As can be seen in the Table 1, the OA&S or LS studies only focused on determining schedules of the orders to minimize the production cost without taking account of the distribution costs and revenue of the orders. While to achieve business goals integrating the production and distribution scheduling is critical (Chen, 2010). Vroblefski et al. (2000) stated that one of the essential costs in distribution is the cost of transportation that is highly dependent on the volume of the orders being transported. In many industries, the Batch Delivery decreases the cost of transportation. Agnetis et al. (2017) studied the batch delivery problem with fixed departure times. In their study, there are  $m$  manufacturers that are modeled as single machines. Gong et al. (2016) considered the flow of products in the iron and steel industry and studied an identical parallel machine scheduling problem with batch deliveries. Yin et al. (2016) investigated a single machine batch delivery scheduling in a make-to-order production system involving two competing agents. In their problem, for each batch, before the processing of the first job, a batch setup time is considered. Rostami et al. (2015) considered single machine scheduling a set of jobs with release times that are to be delivered to a customer or another machine in as batch. Table 2 summarizes the assumptions and features of some of the recently closely related papers of batch

delivery. As can be seen, only two studies, Noroozi et al. (2017) and Noroozi et al. (2018), presented the first study of batch delivery considering simultaneous order acceptance and to the best of researchers' knowledge. In the current study, this is the first time that a coordinated order acceptance, batch delivery, and lot streaming optimization of in a flexible flow shop scheduling has been addressed.

**Table 2**

Summary of the closely related batch delivery

Ref.	OA&S	LS	Production Environment				3PL	Objective			Solution Approach		
			S	P	FS	FFS		Service	Cost	Revenue	DP	B&B	Meta
Rasti-Barzoki et al. (2013)			✓		✓		✓	✓				✓	
(Rasti-Barzoki & Hejazi, 2013)			✓				✓	✓				✓	
(Mazdeh et al., 2013)			✓				✓	✓				✓	
(Mazdeh et al., 2012)			✓				✓					✓	SA
(Ahmadizar & Farhadi, 2015)			✓				✓	✓					ICA
(Assarzadegan & Rasti-Barzoki, 2016)			✓				✓	✓					GA&SA
(Hassanzadeh et al., 2016)					✓		✓	✓					GA&PSO
(Mazdeh et al., 2011)			✓				✓	✓			✓	✓	
(Mazdeh & Rostami, 2014)					✓		✓	✓				✓	
(Rostami et al., 2015)			✓				✓	✓				✓	GA&PSO
(Wan & Zhang, 2014)				✓			✓				✓		
(Yin et al., 2014)			✓				✓	✓			✓		
(Yin et al., 2016)			✓				✓	✓			✓		
(Gao et al., 2015)			✓				✓	✓					
(Noroozi et al., 2017)	✓		✓				✓	✓	✓	✓			GA+PSO
(Noroozi et al., 2018)	✓		✓				✓	✓	✓	✓			GA
<b>This study</b>	✓	✓				✓	✓	✓	✓	✓			GA

In distribution and delivery, as many companies are unable to provide sufficient transportation facilities to deliver the customer orders due to high costs of initial investment, transportation is outsourced to the third-party logistics (3PL) providers in many practical cases (Agnētis et al., 2014; Mehri et al., 2013; Pourghahreman & Qhatari, 2015; Rahchamandi & Fallahi, 2014). Aguezzoul (2014) presented a review on 3PL studies. The aim of this study is to provide a comprehensive mixed integer linear programming to joint order acceptance, lot streaming at a flexible flow shop and batch delivery by considering third-party logistics of an integrated production-distribution scheduling. The objective of the problem is to maximize the total net profit (TNP) by considering the revenues, earliness penalties, tardiness penalties, holding cost, setup time and cost and batch transportation costs. The idea of the considered problem is extracted from the yogurt production and distribution process in the dairy industry. The problem is strongly NP-hard. So, the second aim of this study is to develop the effective solution approach. To do so, a hybrid metaheuristic based on the Genetic Algorithm (GA) is developed. In the proposed algorithm, (1) a heuristic, (2) a local search, and (3) a restart phase are proposed to improve the performance and efficiency of the algorithm. Taguchi experimental design was applied to set the appropriate parameters of the algorithms. The proposed model is solved using commercial software. To evaluate the performance and efficiency of the algorithms, several test problems are generated and the performance of the algorithm is evaluated.

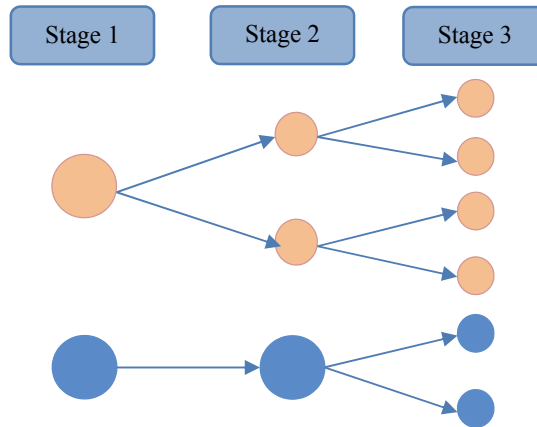
The rest of the paper is organized as follows. In section 2, the novel mixed integer programming model is proposed after describing the problem in detail. Section 3 presents different parts of the developed algorithm. Factors of the problem, data generation, and parameter calibration are described in Section 4, and Finally, Section 5 presents the conclusion.

## 2. Problem statement and mathematical formulation

### 2.1. Problem statement

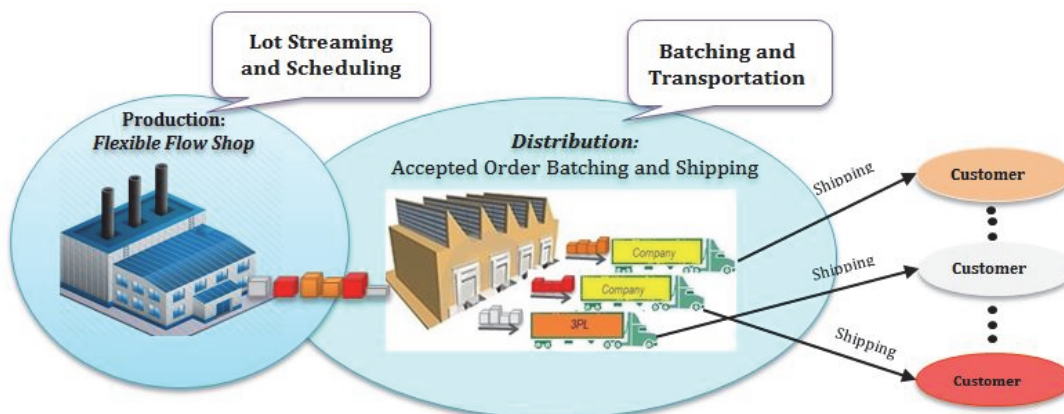
The idea of the considered problem is extracted from the yogurt production and distribution process in the dairy industry. The considered system produces several groups of products. Each group of products has subgroups, namely platform. The platforms have different sizes in which, each customer orders different numbers of some or all of the platforms. In other words, the sizes of the ordered platforms are

non-identical. A pool of orders brings to the company. The production and distribution, depended on the production and distribution capacity and due to the perishable products, must accept a part of orders to produce and distribute. In the lot streaming, the transformed sub-lots between different stages must have the same group. Each sub-lot of a stage belongs to the group of the lot of the pervious stage. In fact, a sub-lot belonging to a group must be transformed to a lot with the same group at next stage. Consider Fig. 1 where each circle is a sub-lot of a stage and the color indicates its group. The group of the sub-lots of each stage is similar to the pervious transformed sub-lot. At the last stage, the sub-lots of product groups are converted into a number of final products, i.e., platforms.



**Fig. 1.** Lot streaming such as the transformed batches has same group

Each sub-lot has the maximum and minimum size due to the machine. The processing time of a sub-lot depends on the products in the sub-lot, i.e., the size of the sub-lot. A setup time and setup cost exist to start processing each sub-lot. For delivery, the accepted orders of the customer have to be placed in the batches. The direct delivery method is used to deliver the batches to the customers. In directing delivery method, a batch cannot contain two or more customers' orders. The size of ordered platforms is non-identical and the total occupied space of all the orders in a batch should not exceed the maximum capacity of the vehicle. The delivery time of each order in the batch is equal to the maximum completion time of the orders in the batch and the transportation time. It is assumed that the company does not have the sufficient number of the vehicles and if it is needed, more vehicles will be hired from the 3PL provider. Each batch has a loading time on the vehicle dependent on the size of the batch. Each delivered order has revenue. If a product is transported later than its completion time, a holding cost is incurred. The orders have to be delivered at most of their upper bound or lower bound of the due window; otherwise, a tardiness or earliness penalty is incurred. In addition, each customer has a maximum allowable due date that if the order is delivered later than this due date, the customer does not receive the orders and the orders have to be returned. In this condition, the company is sustained the high cost.



**Fig. 2.** An overview of the problem

In a coordination manner of the order acceptance, lot streaming at a flexible flow shop and batch delivery, the main objective of this section is to provide a comprehensive mixed integer linear programming model, and to maximize the total net profit (TNP) with consideration of the following points: 1) multiple customers may have numerous orders, 2) The orders occupy various amount of space, 3) the maximum occupied space by the orders in a batch must not be exceed the vehicle capacity, 4) The transportation can be outsourced, and 5) A part of orders may not be accepted. Fig. 2. presents an overview of the considered problem.

## 2.2. Mathematical formulation

The important notations used throughout this paper are defined as follows:

### Indexes list

$k$	The customer number ( $k = 1, \dots, K$ )
$g$	The product number ( $g = 1, \dots, G$ )
$o$	The platform number ( $o = 1, \dots, O_g$ )
$i, i'$	The sub-lot number ( $i, i' = 1, \dots, NLS$ )
$j$	The stage number ( $j = 1, \dots, S$ )
$m$	The machine number ( $j = 1, \dots, M_j$ )
$b$	The batch number ( $b = 1, \dots, NLS$ )
$BigM$	A big positive number

### Parameters list

$Q_{kgo}$	The revenue of the $o$ th platform of the $g$ th group of the $k$ th customer
$w_{kgo}^t$	The tardiness penalty of the $o$ th platform of the $g$ th group of the $k$ th customer
$w_{kgo}^e$	The earliness penalty of the $o$ th platform of the $g$ th group of the $k$ th customer
$w_{ij}^{st}$	The set up cost of the $i$ th sub-lot at the $j$ th stage
$\gamma_k$	The transportation cost of the $k$ th customer using company's vehicle
$\gamma'_k$	The transportation cost of the $k$ th customer using 3PL's vehicle
$HoldC$	The holding cost
$\alpha$	The minimum size of sub-lot
$\beta_j$	The maximum size of sub-lot at the $j$ th stage
$NLS$	The maximum number of sub-lots
$Plt_o$	The weight of the $o$ th platform (gr)
$Dem_{kgo}$	The demand of the $o$ th platform of the $g$ th group of the $k$ th customer
$MxTr$	The maximum allowable tardiness
$V$	The number of company's vehicles
$Cap$	The loading capacity vehicle
$P_j$	The unit processing time at the $j$ th stage
$St_j$	The set up time at the $j$ th stage
$PcT$	The unit loading time
$WiT$	Maximum waiting time between complete time of a sub-lot at last stage and ready time of that for loading and delivery
$[e_{kgo}, d_{kgo}]$	The due window of the $o$ th platform of the $g$ th group of the $k$ th customer

In following, the variables are introduced:

#### Decision variables

To describe the model, first, the sets of decision variables are defined.

$$h_{ij} = \begin{cases} 1, & \text{If the } i\text{th sub-lot of the } j\text{th stage is formed} \\ 0, & \text{Otherwise} \end{cases}$$

$$Z_{iji'j+1} = \begin{cases} 1, & \text{If a part of or complete size of the } i\text{th sub-lot of the } j\text{th stage transform to the} \\ & i'\text{th sub-lot of the } j + 1\text{th stage,} \\ 0, & \text{Otherwise} \end{cases}$$

$$X_{ijg} = \begin{cases} 1, & \text{If the } i\text{th sub-lot of the } j\text{th stage is assigned to the } g\text{th group} \\ 0, & \text{Otherwise} \end{cases}$$

$$X'_{io} = \begin{cases} 1, & \text{If the } i\text{th sub-lot of the last stage is assigned to the } o\text{th platform} \\ 0, & \text{Otherwise} \end{cases}$$

$$Z'_{kgoi} = \begin{cases} 1, & \text{If the order of the } o\text{th platform of the } g\text{th group of the } k\text{th customer is supplied} \\ & \text{from the } i\text{th sub-lot} \\ 0, & \text{Otherwise} \end{cases}$$

$$A_{kgoib} = \begin{cases} 1, & \text{If the order of the } o\text{th platform of the } g\text{th group of the } k\text{th customer supplied} \\ & \text{from sub-lot } i, \text{ is assigned to the } b\text{th batch} \\ 0, & \text{Otherwise} \end{cases}$$

$$u_{bk} = \begin{cases} 1, & \text{If the } b\text{th batch of the } k\text{th customer is transported using the } b\text{th vehicle} \\ 0, & \text{Otherwise} \end{cases}$$

$$Y_{ijm} = \begin{cases} 1, & \text{If the } i\text{th sub-lot of stage } j \text{ is assigned to machine } m \\ 0, & \text{Otherwise} \end{cases}$$

$LS_{ij}$	The size of the $i$ th sub-lot of stage $j$ (if $h_{ij} = 1$ ).
$q_{iji'j+1}$	Determines how much of the $i$ th sub-lot of stage $j$ transform to the $i'$ th sub-lot of the $j + 1$ th stage.
$Su_{tkgoi}$	How much of the order of the $o$ th platform of the $g$ th group of the $k$ th customer is supplied from the $i$ th sub-lot.
$q'_{kgoib}$	assigns the $Su_{kgoi}$ to the batches of the $k$ th customer
$q''_{kgoib}$	assigns the $Su_{kgoi}$ to the batches of the $k$ th customer
$Sr_{ij}$	The production start time of the sub-lot $i$ in stage $j$ .
$C_{ij}$	The production completion time of the sub-lot $i$ in stage $j$ .
$R_{kgoi}$	The ready time of a supplied order for transporting and delivery.
$R'_{bk}$	The ready time of a batch.
$R''_{kgoi}$	Duration time between the ready time of a supplied order for transporting and the ready time of a batch to which the supplied order belongs.
$Tar_{kgoi}$	The tardiness of the supplied order of the $o$ th platform of the $g$ th group of the $k$ th customer.
$Ear_{kgoi}$	The earliness of the supplied order of the $o$ th platform of the $g$ th group of the $k$ th customer.

Using these decision variables, the proposed mixed linear integer programming model of the problem can be described as follows:

*Objective Function:*

$$\begin{aligned}
 Max Z = & \sum_{b=1}^{NLS} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} Q_{go} \frac{q''_{kgoib}}{Plt_{go}} - \sum_{b=1}^{NLS} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} w_{kgo}^t Tar_{kgoib} \\
 & - \sum_{b=1}^{NLS} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} BigM Tar'_{kgoib} - \sum_{b=1}^{NLS} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} w_{kgo}^e Ear_{kgoib} - \sum_{i=1}^{NLS} \sum_{j=1}^S h_{ij} W_{ij}^{st} \\
 & - \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} HoldCR''_{kgoi} - \left( \sum_{b=1}^V \sum_{k=1}^K (\gamma_k u_{bk}) + \sum_{b=V+1}^{NLS} \sum_{k=1}^K (\gamma'_k u_{bk}) \right)
 \end{aligned} \tag{1}$$

Before describing the constraints of the model, the model objective function is presented below separately (maximizing the TNP):

$\sum_{b=1}^{NLS+V} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} Q_{go} \frac{q''_{kgoib}}{Plt_{go}}$	The revenue obtained from accepted order (supplies orders)
$\sum_{b=1}^{NLS+V} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} w_{kgo}^t Tar_{kgoib}$	The tardiness penalties
$\sum_{b=1}^{NLS+V} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} BigM Tar'_{kgoib}$	The returned batches costs
$\sum_{b=1}^{NLS+V} \sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} w_{kgo}^e Ear_{kgoib}$	The earliness penalties
$\sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS} HoldCR''_{kgoi}$	The holding costs
$\sum_{i=1}^{NLS} \sum_{j=1}^S h_{ij} W_{ij}^{st}$	The setup costs
$\sum_{b=1}^V \sum_{k=1}^K (\gamma_k u_{bk}) + \sum_{b=V+1}^{NLS+V} \sum_{k=1}^K (\gamma'_k u_{bk})$	The transportation cost

For simplicity, we describe the constraints in four groups.

*Lot streaming:*

$$LS_{ij} \geq h_{ij} \alpha ; i = 1, \dots, NLS, \quad j = 1, \dots, S \tag{2}$$

$$LS_{ij} \leq h_{ij} \beta_j ; i = 1, \dots, NLS, \quad j = 1, \dots, S \tag{3}$$

$$q_{ij} i'_{j+1} \leq BigM \cdot Z_{ij} i'_{j+1} ; i', i = 1, \dots, NLS, \quad j = 1, \dots, S - 1 \tag{4}$$

$$\sum_{i=1}^{NLS} q_{ij-1} i'_j = LS_{i'_j} ; i' = 1, \dots, NLS, \quad j = 2, \dots, S \tag{5}$$

$$\sum_{i=1}^{NLS} Z_{ij} i'_{j+1} \leq 1 ; i' = 1, \dots, NLS, \quad j = 1, \dots, S - 1 \tag{6}$$

$$\sum_{i'=1}^{NLS} q_{ij} i'_{j+1} = LS_{ij} ; i = 1, \dots, NLS, \quad j = 1, \dots, S - 1 \tag{7}$$

$$\sum_{g=1}^G X_{ijg} = 1 ; i = 1, \dots, NLS, \quad j = 1, \dots, S \quad (8)$$

$$X'_{i'j+1g} \geq X_{ijg} - \text{Big}M(1 - Z_{iji'j+1}); g = 1, \dots, G, i = 1, \dots, NLS, j = 1, \dots, S \quad (9)$$

$$X'_{i'j+1g} \leq X_{ijg} - \text{Big}M(1 - Z_{iji'j+1}); g = 1, \dots, G, i = 1, \dots, NLS, j = 1, \dots, S \quad (10)$$

$$\sum_{g=1}^G \sum_{o=1}^O X'_{io} = 1; i = 1, \dots, NLS \quad (11)$$

$$2Z'_{kgoi} \leq X_{iSg} + X'_{io}; k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (12)$$

$$Su_{kgoi} \leq \text{Big}M \cdot Z'_{kgoi}; k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (13)$$

Constraints (2) and (3) ensure that if the  $i$ th sub-lot of the  $j$ th stage is formed, it must be less than the maximum size and greater than the minimum size. According to the variables  $Z_{iji'j+1}$  and  $q_{iji'j+1}$ , constraint (4) determines whether the  $i$ th sub-lot of the  $j$ th stage is transformed to the  $i'$  sub-lot of the  $j + 1$ th stage, and if yes, how much of this sub-lot is transformed to the next stage. Eq. (5) guarantees the sum of the transformed sub-lots from a stage to sub-lots of the next stage that must be equaled to the size of the sub-lot transformed that was determined according to constraints (2) and (3). Furthermore, according to the technical limitation, two or more sub-lots of the pervious stage must not be transformed to a sub-lot of the current stage. Constraint (6) ensures this limitation; however, a sub-lot of a stage can be transformed to one or more sub-lots of the next stage. Therefore, the sum of the size of the transformed sub-lots from a sub-lot must be equal to the size of that sub-lot. Eq. (7) guarantees this transforming. The transformed sub-lots between different stages must have the same group. To guarantee these conditions, constraint (8) determines the group of the transformed sub-lot and constraints (9) and (10) guarantee a sub-lot of the current stage is transformed to a sub-lot of the next stage with the same group. As mentioned, customers order the goods as platforms such as one hundred of group 1 with platform 2. The goods are produced as platforms in the last stage. Constraint (11) assigns the sub-lots to platforms. Constraints (12) and (13) assign the sub-lots to orders. In other words, these constraints determine a part of the order of customer  $k$ , group  $g$  and platform  $o$  is supplied using sub-lot  $i$ . The considered problem is order acceptance.

#### Order acceptance:

$$\sum_{i=1}^{NLS} \frac{Su_{kgoi}}{Plt_{go}} \leq Dem_{kgo}; k = 1, \dots, K, \quad g = 1, \dots, G, o = 1, \dots, O_g \quad (14)$$

$$\sum_{k=1}^K \sum_{g=1}^G \sum_{o=1}^{O_g} Su_{kgoi} = LS_{iS}; i = 1, \dots, NLS \quad (15)$$

Constraint (14) guarantees the sum of supplied orders that must be less than the weight (demand) of the order. This constraint states a part of an order of a product that can be accepted and supplied, and others are rejected. The sum of supplied orders from a sub-lot must be equal to the size of that sub-lot. Eq. (15) guarantees this supplying.

#### Batching:

$$\sum_{b=1}^{NLS+V} q'_{kgoib} = Su_{kgoi}; k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (16)$$



$$\sum_{k=1}^K u_{bk} \leq 1 ; \quad b = 1, \dots, NLS + V \quad (17)$$

$$\sum_{b=1}^{NLS+V} u_{bk} \leq NLS ; \quad k = 1, \dots, K \quad (18)$$

$$q'_{kgoib} \leq BigM \cdot A_{kgoib} ; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS , \quad b = 1, \dots, NLS + V \quad (19)$$

$$q''_{kgoib} \leq q'_{kgoib} ; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS , \quad b = 1, \dots, NLS + V \quad (20)$$

$$q''_{kgoib} \geq q'_{kgoib} - BigM \cdot (1 - A_{kgoib}); \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS , \quad b = 1, \dots, NLS + V \quad (21)$$

$$A_{kgoib} \leq U_{bk}; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS , \quad b = 1, \dots, NLS + V \quad (22)$$

$$\sum_{g=1}^G \sum_{o=1}^{O_g} \sum_{i=1}^{NLS+V} q''_{kgoib} \leq CapU_{bk}; \quad b = 1, \dots, NLS + V, \quad k = 1, \dots, K \quad (23)$$

The supplied orders of customer  $k$  of group  $g$  of platform  $o$  supplied by  $i$ th sub-lot would be assigned to one or more batches using equation (16). Constraint (17) assigns each batch to a customer and constraint (18) considers a number of vehicles of company or 3PL for a customer. Constraints (19)–(21) assign the supplied orders to the batches. Constraint (22) guarantees the variable  $A_{kgoib}$  is 1, when a batch place in one of the vehicles of company or 3PL provider, i.e.,  $u_{bk} = 1$ . Constraint (23) ensures that the space occupied by the orders allocated to a batch does not exceed the vehicle capacity.

*Scheduling:*

$$\sum_{m=1}^{M_s} Y_{ijm} = 1 ; \quad i = 1, \dots, NLS, \quad j = 1, \dots, S \quad (24)$$

$$Sr_{i't'j} - BigM(Y_{ijm} + Y_{i't'jm} - 2) \geq C_{ij}; \quad i, i' = 1, \dots, NLS, i < i', j = 1, \dots, S \quad (25)$$

$$Sr_{i't'j+1} - BigM(1 - Z_{ij'i'j+1}) \geq C_{ij}; \quad i, i' = 1, \dots, NLS, j = 1, \dots, S - 1 \quad (26)$$

$$Sr_{ij} + LS_{ij}P_j + St_{h_{ij}} \leq C_{ij}; \quad i = 1, \dots, NLS, j = 1, \dots, S \quad (27)$$

$$R_{kgoi} + BigM(1 - Z'_{kgoi}) \geq C_{is}; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (28)$$

$$R_{kgoi} - (C_{is}) \leq WiT + BigM(1 - Z'_{kgoi}); \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (29)$$

To calculate the completion time of a sub-lot, equation (24) assigns each sub-lot to a machine, and constraints (25) and (26) compute the processing start time of a sub-lot in each stage. The processing start time of a batch is maximum of the completion time of the pervious sub-lot on the same machine and the completion time of this sub-lot in the previous stage. According to these constraints, the completion time of a sub-lot is computed by constraint (27). The ready time of the supplied order by sub-lot  $i$  is equal to the completion time of the sub-lot  $i$ , that is computed by constraints (28) and (29).

*Batch delivery:*

$$R'_{bk} + BigM(1 - A_{kgoib}) \geq R_{kgoi} + (PcT \times \frac{q''_{kgoib}}{Plt_{go}}); \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, \quad i = 1, \dots, NLS , \quad b = 1, \dots, NLS + V \quad (30)$$

$$R''_{kgoi} + \text{BigM}(1 - A_{kgoib}) \geq R'_{bk} - (R_{kgoi} + \left( PCT \times \frac{q''_{kgoib}}{Plt_{go}} \right)); \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, \quad (31)$$

$$i = 1, \dots, NLS, \quad b = 1, \dots, NLS + V$$

$$R''_{kgoi} - \text{BigM}(1 - A_{kgoib}) \leq R'_{bk} - (R_{kgoi} + \left( PCT \times \frac{q''_{kgoib}}{Plt_{go}} \right)); \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, \quad (32)$$

$$i = 1, \dots, NLS, \quad b = 1, \dots, NLS + V$$

$$Del_{bk} = R'_{bk} + \tau_k; \quad k = 1, \dots, K, b = 1, \dots, NLS + V \quad (33)$$

When a batch is formed, the ready time of a batch,  $R'_{bk}$ , is equal to the maximum ready time of the orders of the batch plus loading time. Constraint (30) states the ready time of a batch for shipping that is at least equal to the ready time of the order in this batch plus loading time. Constraints (31) and (32) compute the duration time between the ready time of a supplied order for transporting and the ready time of a batch that the supplied order belongs to. The delivery time of a batch is computed using constraint (33). The time spent between shipping time of a batch and ready time of the orders in the batch causes the holding cost.

### Earliness and tardiness:

$$Tar_{kgoib} + \text{BigM}(1 - A_{kgoib}) \geq Del_{bk} - d_{kgo}; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (34)$$

$$Tar'_{kgoib} + \text{BigM}(1 - A_{kgoib}) \geq Del_{bk} - MxTr_k; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (35)$$

$$Ear_{kgoib} + \text{BigM}(1 - A_{kgoib}) \geq Del_{bk} - e_{kgo}; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS \quad (36)$$

Finally, with the delivery time of the orders in batches, constraints (34) and (35), respectively, compute the correct value of the order tardiness if it would be delivered after the upper bound of the due window and if it would be delivered after the maximum allowable due date. Similarly, constraint (36) computes the order earliness.

### Variables:

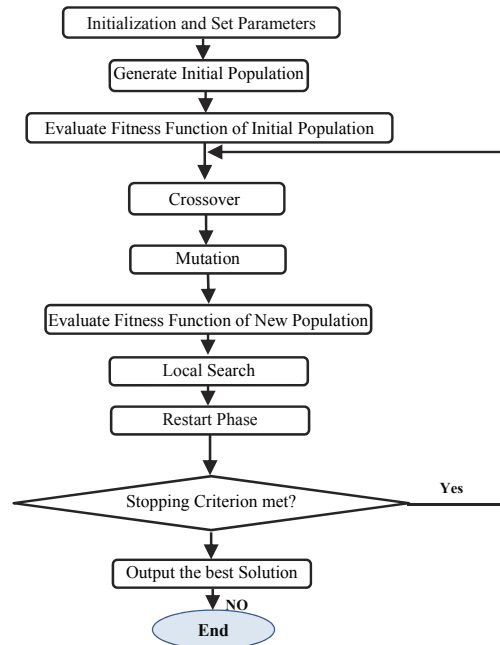
$$LS_{ij}, C_{ij}, q'_{kgoib}, q''_{kgoib}, Su_{kgoi}, R_{kgoib}, R''_{kgoi}, R'_{bk}, Tar_{kgoib}, Tar'_{kgoib}, Ear_{kgoib} \geq 0; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i = 1, \dots, NLS, b = 1, \dots, NLS \quad (37)$$

$$h_{ij}, Z_{ij|j+1}, X_{ijg}, X_{io}, A_{kgoib}, Del_{bk}, Y_{ijm}, Z'_{kgi} \in \{0,1\}; \quad k = 1, \dots, K, g = 1, \dots, G, o = 1, \dots, O_g, i, i' = 1, \dots, NLS, j = 1, \dots, S, m = 1, \dots, M_s \quad (38)$$

Constraints (37)–(38) introduce the decision variables. Batch delivery and the order acceptance problem are a strongly NP-hard problem (Chen, 2010; Herbots et al., 2007). Therefore, the considered problem is more complex than the classical scheduling and is strongly NP-hard.

### 3. Genetic algorithm

Due to computational time constraints, the complete enumeration of the solution space or application of the exact methods is not practical. In this paper, a HGA has been developed, which is an evolutionary algorithm. Fig. 3 indicates the flow chart of the proposed genetic algorithm. In this paper, a straightforward and easy-to-apply is proposed for this purpose. In addition to the genetic operators, i.e. crossover and mutation operators, a local search procedure and a restart phase has been developed to enhance the search mechanism.



**Fig. 3.** Flow Chart of Proposed Genetic Algorithm

**3.1. Encoding scheme**

The encoding schemes are used to present recognizable solutions for the algorithms and computers. Given the complexity of the problem and the decision-making items such as lot streaming, batch delivery and order acceptance, one of the most important challenges in developing the algorithm for the problem is how a solution is encoded by a representation mechanism. The structure of the proposed chromosome for the encoding scheme is as a matrix with  $S + 2$  rows and  $NLS$  columns. The first row is for order acceptance, the rows 2 to  $S + 1$  is for lot streaming, sub-lot scheduling and batch delivery at each stage and the last row is for maximum waiting time between completion time of last stage and ready time to shipping. Fig. 4 shows the matrix. In this chromosome, a Random Keys is placed in each gene.

$D_{11}^1$	$D_{12}^1$	...	$D_{21}^1$	$D_{22}^1$	...	...	$D_{11}^K$	$D_{12}^K$	...	$D_{G0g}^K$
$B_{11}$	$B_{21}$	...								$B_{Max1}$
$B_{12}$	$B_{22}$	...								$B_{Max2}$
	.									.
	.									.
$B_{1S}$	$B_{2S}$	...								$B_{MaxS}$
$Wit1$	$Wit2$	...								$WitS$

**Fig. 4.** Encoding scheme of proposed GA

We describe the details with an example. Consider a demand table presented in Table 3.

**Table 3**  
An example of demands

		Group , Platform					
		g=1 , o=1	g=1 , o=2	g=2 , o=1	g=2 , o=2	g=3 , o=1	g=3 , o=2
Customer	K = 1	200	1000	750	3000	300	700
	K = 2	400	2500	1200	600	400	300

**Step 1.** Make equal intervals with  $\Delta$  size using equation (39):

$$\Delta_g = \frac{1}{\text{SumOfOrder}} = \frac{1}{12} = 0.083; \quad (39)$$

Instead of the numbers smaller than  $\Delta$ , put zero. This means that the corresponding order is rejected. Multiply the random keys of first row in customer demands. The results show the amount of demand that should be produced and delivered. Multiply these demands on the platform weight to obtain the weight of demands. Sum up the weight of customer demands by platform. The results show the production required for each platform (Fig. 5).

**Step 2.** Now, for each platform, consider a number of sub-lots in the stages. To achieve this, first, multiply the demand for each platform in its revenue and then, normalize the resulting numbers. Using formula (1) to each platform, depending on the revenue and the amount of demand, a proportionate number of sub-lots is assigned (see Fig. 6).

	0.616	0.687	0.919	0.871	0.149	0.501	0.885	0.076	0.845	0.217	0.831	0.957
	0.616	0.687	0.919	0.871	0.149	0.501	0.885	< $\Delta_g$	0.845	0.217	0.831	0.957
	×											
	200	1000	750	3000	300	700	400	2500	1200	600	400	300
	=											
Accepted Demand-Number	123	687	689	2612	45	351	354	0.00	1014	130	333	287
	×											
Weight of Platform	200	750	600	150	50	30	200	750	600	150	50	30
	=											
Accepted Demand-Weight	24636	515325	413595	391815	2232	10527	70824	0	608472	19512	16626	8616

Weighted Demand of paltforms		O=1	O=2
	g=1	95,460	515,325
	g=2	1,022,067	411,327
	g=3	18,858	19,143

**Fig. 2.** the production required for each platform according to the example of Table 3

Weighted Demand × Revenue of Paltforms

	O=1	O=2
g=1	3,818,400	61,839,000
g=2	143,089,380	14,396,445
g=3	1,508,640	861,435

Normalize of Weighted Demand × Revenue of Paltforms

	O=1	O=2
g=1	0.0169	0.2742
g=2	0.6345	0.0638
g=3	0.0067	0.0038

Maximum Number of sub-lots = 15

Assigned Sub-lots to the Paltforms

	O=1	O=2
g=1	1.00	3.00
g=2	6.00	1.00
g=3	1.00	1.00

**Fig. 6.** Assigning the sub-lots to the Paltforms

We now should assign demands to the sub-lots of the last stage. For each platform, it should be make equal intervals with the  $\Delta_{go}$  length that a number is assigned to the sub-lots in ascending order. These intervals determine which part of the demand is assigned to which sub-lot. The random key of each demand is in the interval, the corresponding sub-lot will produce the corresponding demand (see Fig. 7).

$$\Delta_{go} = \frac{1}{\text{number of assigned batches}}; \quad g = 1, \dots, G; \quad o = 1, \dots, O_g \tag{40}$$

Platform(g,o)	(1,1)	(1,2)	(1,2)	(1,2)	(2,1)	(2,1)	(2,1)	(2,1)	(2,1)	(2,1)	(2,2)	(3,1)	(3,2)
Sub-lots of last stage	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	B <sub>8</sub>	B <sub>9</sub>	B <sub>10</sub>	B <sub>11</sub>	B <sub>12</sub>	B <sub>13</sub>
Interval	[0,1]	[0,0.33]	[0.33,0.66]	[0.66,1]	[0,0.17]	[0.17,0.34]	[0.34,0.51]	[0.51,0.68]	[0.68,0.85]	[0.85,1]	[0,1]	[0,1]	[0,1]
	0.616	0.68	0.919	0.571	0.149	0.501	0.885	0.076	0.845	0.217	0.831	0.957	
Sub-lot of last stage	B <sub>1</sub>	B <sub>4</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>5</sub>	B <sub>7</sub>	B <sub>10</sub>	B <sub>5</sub>	B <sub>9</sub>	B <sub>11</sub>	B <sub>12</sub>	B <sub>13</sub>	

**Fig. 7.** Assigning demands to the sub-lots of last stage

By assigning the demands to the sub-lots of the last stage, the number of sub-lots for the previous stage of each platform is determined. Similar to the assigning demands to the final stage, the sub-lots of each stage are allocated to the previous stage using the random keys. If the volume of a sub-lot exceeds the capacity of a machine, another sub-lot is selected randomly so that its size does not exceed the capacity of the sub-lot.

**Step 3.** To schedule and compute the completion time of each sub-lot at each stage, the sub-lots must first be assigned to the machines of each stage. Then, the order of the sequence of sub-lots on the machines will be according to the descending order of their random keys. In other words, the sub-lots with smaller random keys are processed earlier. At this stage, the waiting time between the completion time of a sub-lot in the last stage and the ready time for delivery is also to be computed. To do this, the corresponding random key at each stage in the row  $S + 2$  of the matrix is multiplied by the time of  $Wit$ .

**Step 4.** For batching, we assign the sub-lots of the last stage to the customers using heuristic **H**.

**Heuristic H:** For customer  $k$ , the first supplied order in the schedule is placed in batch 1 of the customer. For the second supplied order, if the size of the supplied order in the schedule is not larger than the remaining capacity of the batch, the supplied order is placed in the batch. Otherwise, if the size is larger than the remaining capacity of the batch, the size of the third supplied order is checked, and this procedure continues until no supplied order can be placed in the batch and then this batch is closed. With respect to un-batched supplied orders, this process is repeated by a new batch. This process is repeated for all supplied orders of the customers. It should be noted that the vehicle’s shipping time is equal to the maximum completion time between the allocated batches plus the loading time. Based on this time, tardiness, earliness, maximum tardiness and holding time are computed.

### 3.2. Selection mechanism

Selection is a genetic mechanism that chooses the solutions as parents for reproduction and determines which solutions in the current population should be selected to produce the next generation of population. In addition to evolutionary operators, i.e. crossover and mutation, the selection mechanism could be effective in the diversity control. There are two common selection mechanisms: (1) Tournament selection: a tournament among solutions is conducted and the best chromosomes are selected to produce new generation and (2) Roulette wheel: a solution with better fitness will have a greater probability to be selected as a parent. In this paper, with preliminary experiments, tournament selection had a better performance than roulette wheel strategy.

### 3.3. Genetic operators

#### 3.3.1. Crossover

To make a partial change, crossover operator generates one or two offspring by swapping parts of parents’ genes. In this paper, three common crossover operators inspired by (Noroozi & Mokhtari, 2015) are employed.

- *One-point crossover*: a random cut-point is generated and between two parents, the gens before that point are swapped.
- *Two-point crossover*: two random cut-points are generated and the chromosome is divided into three parts. The gens in the second part (middle part) are swapped between two parents.
- *Uniform crossover*: each position of parents is compared with one another and their elements are swapped with a probability. A random vector between 0 and 1 is generated such that the vector size is equal to the number of orders. For each member, if the random value is less than the probability value, the gen of the first parent is copied to the offspring; otherwise, the member of the second parent is copied.

In each iteration, a random number between 0 and 1 is generated and

- If random number is  $\leq P_{OP}$ , then apply *one-point crossover*.
- If  $P_{OP} \leq$  random number is  $\leq P_{TP}$ , then apply *two-point crossover*.
- If  $P_{TP} \leq$  random number is  $\leq P_{UC}$ , then apply *uniform crossover*

### 3.3.2. Mutation

Mutation operator is always used to address the local optimality of designing intelligent optimization algorithms. The probability of mutation is defined as the probability that an offspring of crossover operator undergoes during a mutation operation. Three well-known mutations presented in the literature of scheduling the problems are used in this study, which are:

- *Interchange mutation*: two randomly-chosen genes of the chromosome are interchanged.
- *Inversion mutation*: the genes between two randomly-chosen cut-points are reversed.
- *Insertion mutation*: a gene is chosen randomly and is inserted in another position of the chromosome.

In each iteration, a random number between 0 and 1 is generated and

- If random number is  $\leq P_{Int}$ , then apply *interchange mutation*.
- If  $P_{Int} \leq$  random number  $\leq P_{Inv}$ , then apply *inversion mutation*.
- If  $P_{Inv} \leq$  random number  $\leq P_{Ins}$ , then apply *insertion mutation*.

### 3.4. Local search

Local search performs a quick exploration around the solution to locate the algorithm in a better neighborhood of the current solution (Duarte et al., 2017). In this study, the local search is applied to the best solution of each generation. If the new chromosome results in a better objective function, the current chromosome is replaced by the new chromosome. The proposed local search focuses on improving accepting the orders. The proposed local search process of this paper is as follows:

1. Calculate the average of the first row of the random keys of matrix, which is the acceptance or rejection of order,
2. Instead of the numbers less than this average, put the average,
3. If the TNP of the new solution is better, replace this solution.

### 3.5. Restart Phase

In some cases, the population diversity of an algorithm is low to trap of a local optimum. To overcome this problem, the Restart Phase mechanism has been proposed to regenerate the new solution. In this paper, if the TNP is not promoted for more than a pre-specified number of

generations (*RestartPhaseIteration*), the restart phase performs to regenerate the population. The proposed restart phase process of this paper is as follows:

1. Sort the numbers in the descending order of the TNP, so that the solution with a better TNP in the first rows of the matrix and those with a worse TNP are placed at the last levels of the matrix.
2. Skip the first *RegenerateRestartPhaserate%* of individuals from the sorted list (the best individuals).
3. The remaining  $1 - \text{RegenerateRestartPhaserate}\%$  of individuals is replaced using applying a crossover with one of the first *RegenerateRestartPhaserate %* best individuals.
4. If the solution is better, replace this solution.

### 3.6. Parameter setting

Generally, the effectiveness of meta-heuristic algorithms depends on correct selection of the parameters. In this paper, the Taguchi Experimental Design is used to investigate the behavior of suggested algorithms with different levels of parameters and to find the most suitable level. The parameters of the algorithms and their levels are depicted in the first two columns of Table 4. By selecting the maximum amount of S/N ratios, the optimum levels of parameters are bold in column *S/NL* of Table 4.

**Table 4**  
The levels and best level obtained for GA

Factors	Type	S/NL			
		GA	GA-LS	GA-Rst	GA-LS-Rst
Population size	A(0) — $\frac{NLS}{2}$	123.2094	124.3669	124.3924	125.0020
	A(1) — <i>NLS</i>	123.3615	124.6541	124.1547	125.0516
	A(2) — $2 \times NLS$	<b>123.4516</b>	<b>124.7451</b>	<b>124.6542</b>	<b>125.1547</b>
Crossover operator	B(0) — $P_{OP} = 0.33, P_{TP} = 0.66, P_{UC} = 1$	123.1264	124.3169	<b>124.9846</b>	125.2305
	B(0) — $P_{OP} = 0.25, P_{TP} = 0.50, P_{UC} = 1$	<b>123.9533</b>	124.4815	124.1547	125.0651
	B(0) — $P_{OP} = 0.50, P_{TP} = 0.75, P_{UC} = 1$	123.1256	<b>124.9421</b>	124.6235	<b>125.5541</b>
	B(0) — $P_{OP} = 0.25, P_{TP} = 0.75, P_{UC} = 1$	123.9513	124.1574	124.1674	125.0361
Probability of mutation	C(0) — 0.15	<b>123.9542</b>	124.1658	124.1235	<b>125.9851</b>
	C(1) — 0.20	123.3651	124.1238	<b>124.9564</b>	125.0614
	C(2) — 0.30	123.6431	<b>124.4982</b>	124.1265	125.1358
Mutation rate	D(0) — $P_{mv} = 0.33, P_{int} = 0.66, P_{ins} = 1$	123.6512	124.1687	124.7456	<b>125.9654</b>
	D(1) — $P_{mv} = 0.25, P_{int} = 0.50, P_{ins} = 1$	<b>123.9561</b>	124.1542	124.6543	125.2653
	D(2) — $P_{mv} = 0.50, P_{int} = 0.75, P_{ins} = 1$	123.2654	<b>124.9534</b>	124.1534	125.4651
	D(3) — $P_{mv} = 0.25, P_{int} = 0.75, P_{ins} = 1$	123.7516	124.4618	<b>124.8456</b>	125.0984
Stop criterion	E(0) — $\frac{NLS}{2}$	<b>123.6532</b>	124.6348	124.1265	125.1258
	E(1) — <i>NLS</i>	123.4516	124.4126	124.4512	125.1495
	E(2) — $2 \times NLS$	123.2314	<b>124.7451</b>	<b>124.7845</b>	<b>125.6514</b>
The number of generation	F(0) — <i>NLS</i>	123.2654	<b>124.2365</b>	<b>124.6534</b>	125.2166
	F(1) — $2 \times NLS$	<b>123.7461</b>	124.1674	124.1258	125.1251
	F(2) — $3 \times NLS$	123.6514	124.1652	124.4618	<b>125.2416</b>
RestartPhaseIteration	G(0) — $0.5 \times NLS$			124.6158	125.0915
	G(1) — <i>NLS</i>			124.4861	<b>125.1694</b>
	G(2) — $2 \times NLS$			<b>124.7915</b>	125.1673
RegenerateRestart Phaserate	H(0) — $0.2 \times \text{Population size}$			124.5371	125.7563
	H(1) — $0.4 \times \text{Population size}$			<b>124.8786</b>	<b>125.8165</b>
	H(2) — $0.6 \times \text{Population size}$			124.8761	125.6286

## 4. Computational results

### 4.1. Instances and parameters

To evaluate the performance of the proposed algorithm, the paper considers a production and distribution, producing and distributing three groups of product with two platforms, e.g. six platforms. Considering previous studies such as (Nobibon & Leus, 2011), five small datasets and six large datasets created. To determine the number of customers and the number of stages in large size problems, 6

combinations from three levels for each number of customer, i.e.  $K = \{3, 15, 10\}$ , and three levels for each number of stages, i.e.  $S = \{3, 5, 10\}$ , are considered as i.e.  $K \times S = \{3 \times 3, 3 \times 5, 3 \times 10, 5 \times 3, 5 \times 5, 5 \times 10, \}$ . Moreover, for small size problems, 5 combinations from  $K = \{1, 2\}$  and  $S = \{2, 3, 4\}$  are considered, i.e.  $K \times S = \{1 \times 2, 2 \times 2, 1 \times 3, 2 \times 3, 2 \times 4, \}$ . In this paper, the data are generated from a uniform discrete distribution defined in terms of intervals as follows:

- The revenue on [50 , 150]
- The number of machines per stage [1 , 5]
- The unit processing time on [0.001 , 0.001]
- The set up time on [1 , 3]
- The transportation time on [100 , 200]
- The set up cost on [100 , 300]
- The tardiness cost on [1 , 5]
- The earliness cost on [1 , 3]
- The transportation cost by company on [5 , 15]

The transportation cost by 3PL is 1.1 times than the cost of the transportation cost by company. The penalty of delivering a batch after the upper bound of due window is considered a very large positive number. The vehicle capacity is considered to be 13000000 for all the test problems. The maximum and minimum size of the sub-lot is considered to be 600000 and 10000, respectively. To compute the due date for each order, we get the average weighted platform weight (*MidLot*). Then we obtain the average number of batches required using the equation (41). In this equation, as all the orders are not accepted, the coefficient *OA* is considered as a percentage of all orders that are accepted. Utilizing the primary experiments, *OA* is considered to be 0.8.

$$NLS = \sum_{g=1}^G \sum_{o=1}^{O_g} (GPlt_{go} \times OA) / MidLot . \quad (41)$$

Additionally, using the equations, the average processing time of each sub-lot each stage and the average production time are calculated, respectively:

$$\bar{P}_s = (P_j \times MidLot \times NLS) / M_j \quad (42)$$

$$P = \frac{\sum_{s=1}^S \bar{P}_s}{S} \quad (43)$$

Then, using these averages and the transportation time of the customer, the upper bound and lower bound of the due window generation interval are achieved, for each customer using the following equation set:

$$\begin{aligned} Qq_k &= P + \tau_k \\ e_{11k} &= q_{11} \times Qq_k \\ e_{12k} &= q_{12} \times Qq_k \\ d_{21k} &= q_{21} \times Qq_k \\ d_{22k} &= q_{22} \times Qq_k \end{aligned} \quad (44)$$

As a result of the above-mentioned calculations, the due window of each customer is generated from the following distribution:

$$e_k \sim [Floor\ of\ e_{11k}, Round\ of\ e_{12k}] \quad (45)$$

$$d_k \sim [Floor\ of\ d_{21k}, Round\ of\ d_{22k}] \quad (46)$$



Furthermore, for each combination of large size problems, four sample problems are created (24 in total). To obtain more reliability, each problem is executed ten times. Therefore, 240 executions are present for each algorithm to solve the model. For the proposed algorithms, the stop criteria are as follows:

1. Reach a specified number of generations.
2. No change in the TNP with the certain number of repetitions.

All the algorithms are implemented using C# programming language (visual studio 2013) on a computer with a 2.6GHz CPU and a 256Mb RAM.

#### 4.2. Algorithm performance

In order to verify the proposed model and evaluate the performance of the algorithm against the exact solution, the commercial solver GAMS (Solver Cplex) is used to solve the small instances of the problems (Table 5). The first column represents the characteristics of the datasets. The *TNP* and *Time* columns show the TNP and CPU time (millisecond) of the algorithm. As can be seen in Table 5, the proposed algorithms can obtain the solutions near to the optimum solution obtained by Cplex and a reasonable time in comparison with Cplex.

**Table 5**  
Comparison of algorithms in small instances

Problem(K × S)	Cplex		GA		GA-LS		GA-Rst		GA-LS-Rst	
	TNP	Time*	TNP	Time*	TNP	Time*	TNP	Time*	TNP	Time*
1 × 2	254873	3000	248937	<1000	252846	<1000	253160	<1000	253956	<1000
2 × 2	687176	<1000	681918	<1000	682716	<1000	683715	<1000	683841	<1000
1 × 3	262036	5,003,000	260167	34,161	260561	32,451	260990	19,514	260990	20,863
2 × 3	604306	3,002,000	603951	863,715	603514	875,261	604008	785,516	604059	829,722
2 × 4	624468	1,526,000	615861	935,634	615892	921,532	615991	861,472	616002	888,980

\*The time unit is millisecond.

The effect of restart phase and local search on the performance of the suggested GA is investigated in this study. As the scale of objective functions in each instance and run is different, the relative percent deviation (*RPD*) is calculated for the problems. *RPD* is a measure that is mostly used in the literature and is defined as follows:

$$RPD = \frac{Max_{sol} - Alg_{sol}}{Max_{sol}}, \quad (47)$$

where  $Alg_{sol}$  is the solution of the algorithm, and  $Max_{sol}$  is the maximum value of the solutions. In this measure, the lowest *RPD* is selected as the best algorithm. **Table** shows the results, in which, GA is the algorithm without the restart phase and local search, GA-LS is the algorithm with the local search, GA-Rst is the algorithm with the restart phase and without local search, and GA-LS-Rst is the algorithm with the restart phase and local search.

**Table 6**  
Average RPD for algorithms

Problem (K × S)	GA	GA-LS	GA-Rst	GA-LS-Rst
3 × 3	0.2165	0.0908	0.1008	<b>0.0159</b>
5 × 3	0.1998	0.0768	0.0787	<b>0.0184</b>
10 × 3	0.2021	0.0779	0.0750	<b>0.0077</b>
3 × 5	0.2077	0.0939	0.0765	<b>0.0206</b>
5 × 5	0.2079	0.0885	0.0798	<b>0.0153</b>
10 × 5	0.1969	0.0773	0.0779	<b>0.0111</b>
<b>Average</b>	0.2051	0.0842	0.0814	<b>0.0148</b>

As Table 6 shows, GA-LS-Rst considerably outperforms the GA, GA-LS and GA-Rst. Furthermore, Fig. 8 indicates that the restart phase yields good efficacy on GA and the proposed local search performs a good exploration around the solution and helps the algorithm to be located in a better neighborhood. This shows that the local search searching the neighborhood of the best solution and the restart phase, and generating more diversity on the algorithm, has a good effect on the algorithm performance.

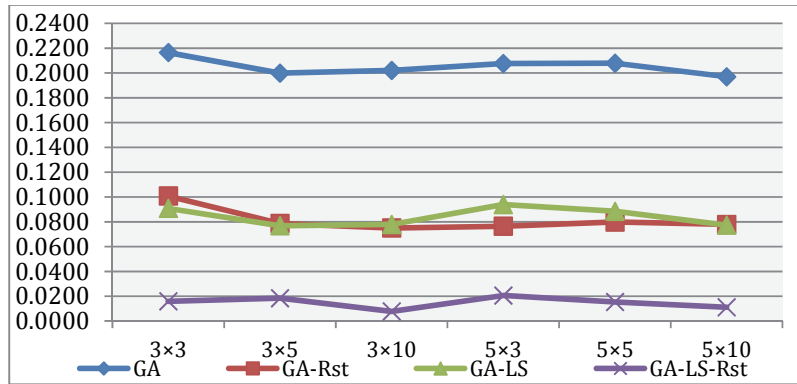


Fig. 3. Interaction between algorithm performance (RPD) and size of problems

The robustness of the evolutionary algorithms is an important measure to assess the reliability of such random search techniques. The robust term was defined as “an adjective referring to the capacity of withstanding vague approximations and/or zones of ignorance in order to prevent undesirable impacts, notably the degradation of the properties to be maintained”. A robust algorithm can be defined as a solution searching method providing consistent results in multiple runs, which are performed using different inputs that correspond to the uncertain parameters of the algorithm. Furthermore, analysis of the robustness of algorithms in terms of standard deviation will be performed in this section. The trend of variation of standard deviations for large size problems obtained by the algorithms is shown in Fig. 9. As shown in the mentioned figure, it seems that the algorithms have similar performance due to the size of problems. Moreover, there is no significant difference among the algorithms in this regard. Furthermore, GA-LS-Rst has the minimum amount of possible standard deviation, which results in different amounts of initial starting points.

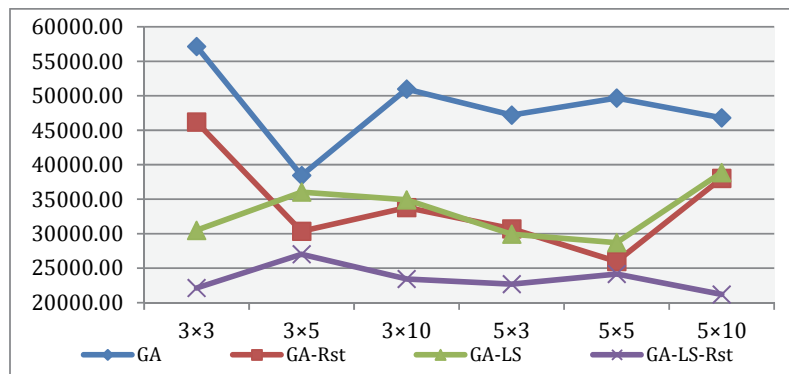


Fig. 9. Comparison of the average of standard deviation of the algorithms

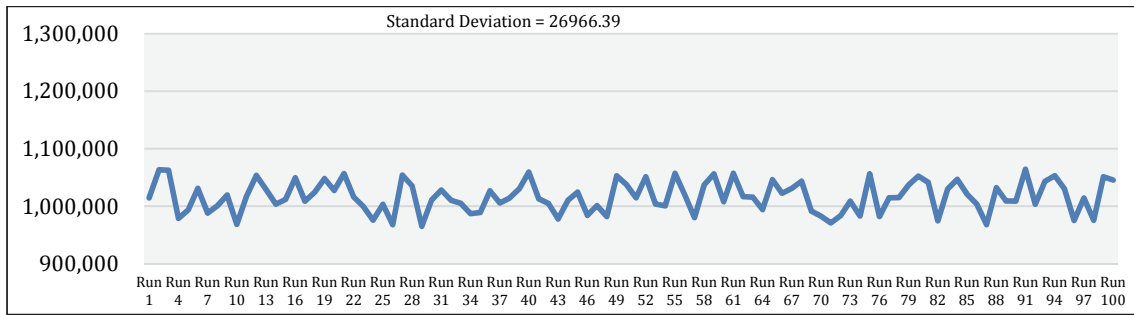
As a further analysis, the test problem  $10 \times 5$  is run 100 times, and coefficients of variation of the algorithms are presented in Table 7.

**Table 7**

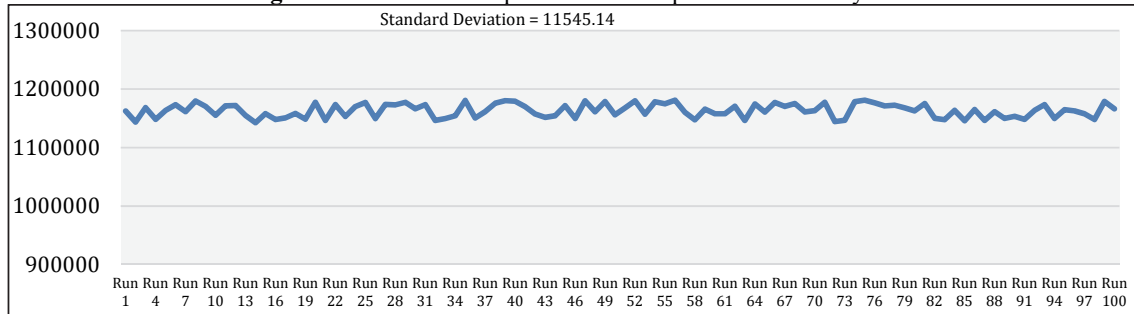
Coefficient of variation of the algorithms

Algorithm	GA	GA-Rst	GA-LS	GA-LS-Rst
Coefficient of Variation	5.38	2.29	2.28	1.98

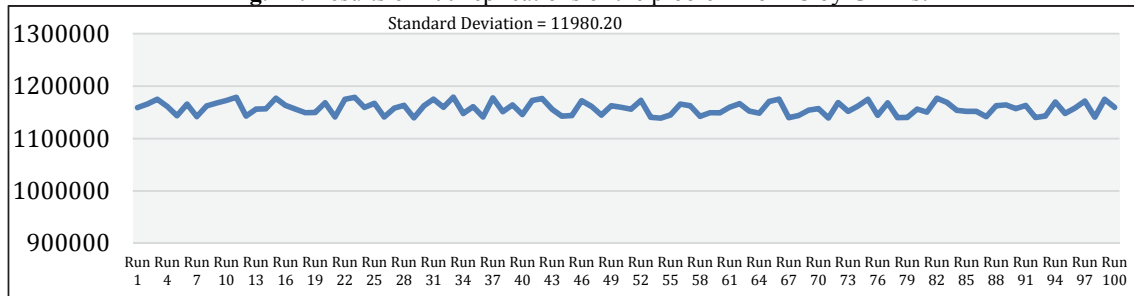
Moreover, the results are presented in Fig. 10 to Fig. 13. As the figures indicate all of the algorithms approximately show similar performance.



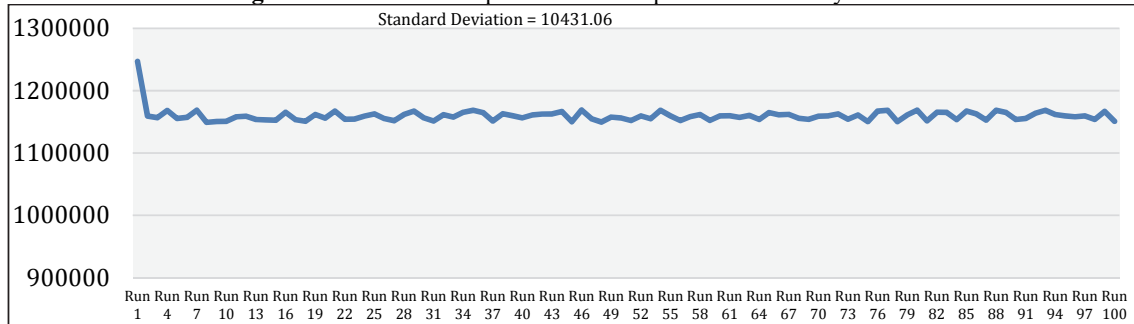
**Fig. 10.** Results of 100 replications of the problem  $10 \times 5$  by GA



**Fig. 11.** Results of 100 replications of the problem  $10 \times 5$  by GA-Rst

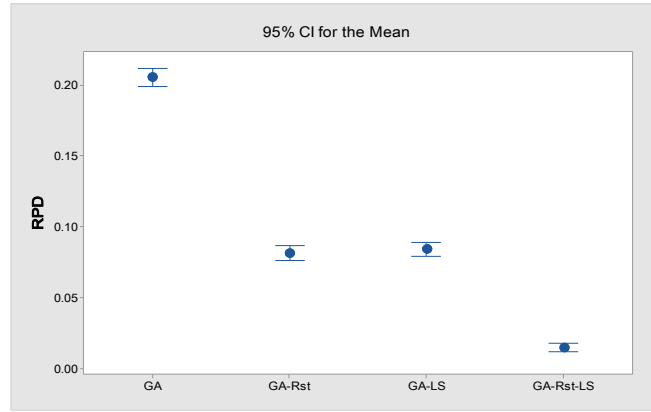


**Fig. 12.** Results of 100 replications of the problem  $10 \times 5$  by GA-LS



**Fig. 13.** Results of 100 replications of the problem  $10 \times 5$  by GA-LS-Rst

Furthermore, to verify the statistical validity of the results, an analysis of variance (ANOVA) is performed to accurately analyze the results. The results show that there is a statistically significant difference between performances of the algorithms. The means plot and LSD intervals of the algorithms at the 95% confidence level are shown in Fig. 14. No overlap was observed between them.



**Fig. 14.** Means plot and LSD intervals for the algorithms

## 5. Conclusion

In this paper, inspired of the yogurt production and distribution process, a novel viewpoint of the scheduling decisions was proposed, which included production scheduling, variable and discrete lot streaming of accepted orders in a flexible flow shop, allocating the batches to the vehicles of the company or 3PL provider and batch delivery.

In this problem, a pool of orders with different groups and platforms is received by the company, where a subset of them can be accepted and the remaining ones should be rejected. Acceptance or rejection is based on maximizing the total net profit according to the revenue, transportation cost, tardiness cost, earliness cost, holding cost and set up cost of orders. The numbers of the vehicles of the company is limited and a Third-Party Logistics provider can deliver the batches to multiple customers. The orders have different sizes and the capacity of vehicles is limited. The main contributions and conclusions of the present study are summarized as follows:

- This study for the first time formulated a coordinated order acceptance, variable and discrete sub-lots of lot streaming in a flexible flow shop, batch delivery and used the 3PL provider scheduling optimization model in an integrated production and distribution system.
- A new mixed integer linear programming was proposed.
- An efficient hybrid genetic algorithm was developed for large-scale problems.
- According to the real properties of the problem, an applicable encoding scheme is proposed to present recognizable solutions.
- A local search was proposed to explore and locate the algorithm in a better neighborhood.
- To overcome trapping of a local optimum and an appropriate population diversity of the algorithm, a restart Phase mechanism has been proposed to regenerate the new solution.
- The appropriate values of the parameters of the algorithms were set with application of Taguchi Experimental Design, and random instances were generated to evaluate the performance of the algorithms.
- The effect of the restart phase and the local search on the performance of the algorithm was investigated.
- Using a commercial solver, the developed model was verified and the performance of algorithm against the exact solution was evaluated.
- The attained results showed the appropriate performance of the proposed HGA.

The researchers of the present study believe that real application of this method, both technically and economically, would be feasible and affordable due to simplicity of the proposed model, heuristics,

local search and algorithm. Furthermore, this study can be developed and applied for other industry such as Automotive industry.

## Acknowledgement

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

## References

- Agnētis, A., Aloulou, M. A., & Fu, L.-L. (2014). Coordination of production and interstage batch delivery with outsourced distribution. *European Journal of Operational Research*, 238(1), 130-142.
- Agnētis, 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, 6193-6206).
- Aguezoul, A. (2014). Third-party logistics selection problem: A literature review on criteria and methods. *Omega*, 49, 69-78.
- 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.
- Ahonen, H., & de Alvarenga, A. G. (2017). Scheduling flexible flow shop with recirculation and machine sequence-dependent processing times: formulation and solution procedures. *The International Journal of Advanced Manufacturing Technology*, 89(1-4), 765-777.
- Assarzadegan, P., & Rasti-Barzoki, M. (2016). Minimizing sum of the due date assignment costs, maximum tardiness and distribution costs in a supply chain scheduling problem. *Applied Soft Computing*, 47, 343-356.
- Bożek, A., & Werner, F. (2017). Flexible job shop scheduling with lot streaming and subplot size optimisation. *International Journal of Production Research*, 1-21.
- Cesaret, B., Oğuz, C., & Salman, F. S. (2012). A tabu search algorithm for order acceptance and scheduling. *Computers & Operations Research*, 39(6), 1197-1205.
- Chaurasia, S. N., & Singh, A. (2016). Hybrid evolutionary approaches for the single machine order acceptance and scheduling problem. *Applied Soft Computing*.
- Chaurasia, S. N., & Singh, A. (2017). Hybrid evolutionary approaches for the single machine order acceptance and scheduling problem. *Applied Soft Computing*, 52, 725-747.
- Chen, C., Yang, Z., Tan, Y., & He, R. (2014). Diversity controlling genetic algorithm for order acceptance and scheduling problem. *Mathematical Problems in Engineering*, 2014.
- Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: review and extensions. *Operations Research*, 58(1), 130-148.
- Cheng, M., Mukherjee, N., & Sarin, S. (2013). A review of lot streaming. *International Journal of Production Research*, 51(23-24), 7023-7046.
- Cheng, M., Sarin, S. C., & Singh, S. (2016). Two-stage, single-lot, lot streaming problem for a 1+ 2 hybrid flow shop. *Journal of Global Optimization*, 66(2), 263-290.
- Duarte, M., Cepeda, J., González-Neira, E., Barrera, D., Cortés, V., & Rey, G. (2017). Performance evaluation of a GRASP-based approach for stochastic scheduling problems. *Uncertain Supply Chain Management*, 5(4), 359-368.
- Emami, S., Sabbagh, M., & Moslehi, G. (2016). A Lagrangian relaxation algorithm for order acceptance and scheduling problem: a globalised robust optimisation approach. *International Journal of Computer Integrated Manufacturing*, 29(5), 535-560.
- Esmailbeigi, R., Charkhgard, P., & Charkhgard, H. (2016). Order acceptance and scheduling problems in two-machine flow shops: new mixed integer programming formulations. *European Journal of Operational Research*, 251(2), 419-431.

- Gao, S., Qi, L., & Lei, L. (2015). Integrated batch production and distribution scheduling with limited vehicle capacity. *International Journal of Production Economics*, 160, 13-25.
- Gong, H., Tang, L., & Leung, J. Y. (2016). Parallel machine scheduling with batch deliveries to minimize total flow time and delivery cost. *Naval Research Logistics (NRL)*, 63(6), 492-502.
- Han, Y., Gong, D., Jin, Y., & Pan, Q.-k. (2016). Evolutionary multi-objective blocking lot-streaming flow shop scheduling with interval processing time. *Applied Soft Computing*, 42, 229-245.
- Hassanzadeh, A., Rasti-Barzoki, M., & Khosroshahi, H. (2016). Two new meta-heuristics for a bi-objective supply chain scheduling problem in flow-shop environment. *Applied Soft Computing*, 49, 335-351.
- Herbots, J., Herroelen, W., & Leus, R. (2007). Dynamic order acceptance and capacity planning on a single bottleneck resource. *Naval Research Logistics (NRL)*, 54(8), 874-889.
- Lalitha, J. L., Mohan, N., & Pillai, V. M. (2017). Lot streaming in  $[N-1](1)+N$  (m) hybrid flow shop. *Journal of Manufacturing Systems*, 44, 12-21.
- Lei, D., & Guo, X. (2015). A parallel neighborhood search for order acceptance and scheduling in flow shop environment. *International Journal of Production Economics*, 165, 12-18.
- Mazdeh, M. M., Haddadm, H., & Ghanbari, P. (2012). Solving a single machine stochastic scheduling problem using a branch and bound algorithm and simulated annealing. *International Journal of Management Science and Engineering Management*, 7(2), 110-118.
- Mazdeh, M. M., & Rostami, M. (2014). A branch-and-bound algorithm for two-machine flow-shop scheduling problems with batch delivery costs. *International Journal of Systems Science: Operations & Logistics*, 1(2), 94-104.
- Mazdeh, M. M., Rostami, M., & Namaki, M. H. (2013). Minimizing maximum tardiness and delivery costs in a batched delivery system. *Computers & Industrial Engineering*, 66(4), 675-682.
- Mazdeh, M. M., Shashaani, S., Ashouri, A., & Hindi, K. S. (2011). Single-machine batch scheduling minimizing weighted flow times and delivery costs. *Applied Mathematical Modelling*, 35(1), 563-570.
- Mehri, S., Roghanian, E., & Khodadadzadeh, T. (2013). A methodology for outsourcing resources in reverse logistics using fuzzy TOPSIS and fuzzy linear programming. *Uncertain Supply Chain Management*, 1(2), 107-114.
- Mukherjee, N. J., Sarin, S. C., & Singh, S. (2017). Lot streaming in the presence of learning in subplot-attached setup times and processing times. *International Journal of Production Research*, 55(6), 1623-1639.
- Nejati, M., Mahdavi, I., Hassanzadeh, R., & Mahdavi-Amiri, N. (2016). Lot streaming in a two-stage assembly hybrid flow shop scheduling problem with a work shift constraint. *Journal of Industrial and Production Engineering*, 33(7), 459-471.
- Nguyen, S. (2016). A learning and optimizing system for order acceptance and scheduling. *The International Journal of Advanced Manufacturing Technology*, 1-16.
- Nobibon, F. T., & Leus, R. (2011). Exact algorithms for a generalization of the order acceptance and scheduling problem in a single-machine environment. *Computers & Operations Research*, 38(1), 367-378.
- Noroozi, A., Mazdeh, M. M., Heydari, M., & Rasti-Barzoki, M. (2018). Coordinating order acceptance and integrated production-distribution scheduling with batch delivery considering Third Party Logistics distribution. *Journal of Manufacturing Systems*, 46, 29-45.
- Noroozi, A., Mazdeh, M. M., Noghondarian, K., Rasti-Barzoki, M., & Heydari, M. (2017). Evolutionary computation algorithms to coordinating order acceptance and batch delivery for an integrated supply chain scheduling. *Computational and Applied Mathematics*, 1-51.
- Noroozi, A., & Mokhtari, H. (2015). Scheduling of printed circuit board (PCB) assembly systems with heterogeneous processors using simulation-based intelligent optimization methods. *Neural Computing and Applications*, 26(4), 857-873.
- Ou, J., & Zhong, X. (2016). Order acceptance and scheduling with consideration of service level. *Annals of Operations Research*, 1-19.

- Pourghahreman, N., & Qhatari, A. (2015). Supplier selection in an agent based pharmaceutical supply chain: An application of TOPSIS and PROMETHEE II. *Uncertain Supply Chain Management*, 3(3), 231-240.
- Rahchamandi, E., & Fallahi, K. (2014). An investigation on logistics outsourcing on exports of minerals goods. *Uncertain Supply Chain Management*, 2(3), 163-166.
- Rasti-Barzoki, M., & Hejazi, S. R. (2013). Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains. *European Journal of Operational Research*, 228(2), 345-357.
- Rasti-Barzoki, M., Hejazi, S. R., & Mazdeh, M. M. (2013). A branch and bound algorithm to minimize the total weighed number of tardy jobs and delivery costs. *Applied Mathematical Modelling*, 37(7), 4924-4937.
- Reisi-Nafchi, M., & Moslehi, G. (2015a). A hybrid genetic and linear programming algorithm for two-agent order acceptance and scheduling problem. *Applied Soft Computing*, 33, 37-47.
- Reisi-Nafchi, M., & Moslehi, G. (2015b). Integrating two-agent scheduling and order acceptance problems to maximise total revenue by bounding each agent penalty function. *International Journal of Services and Operations Management*, 20(3), 358-384.
- Rostami, M., Kheirandish, O., & Ansari, N. (2015). Minimizing maximum tardiness and delivery costs with batch delivery and job release times. *Applied Mathematical Modelling*, 39(16), 4909-4927.
- Sang, H.-Y., Pan, Q.-K., Duan, P.-Y., & Li, J.-Q. (2015). An effective discrete invasive weed optimization algorithm for lot-streaming flowshop scheduling problems. *Journal of Intelligent Manufacturing*, 1-13.
- Silva, Y. L. T., Subramanian, A., & Pessoa, A. A. (2018). Exact and heuristic algorithms for order acceptance and scheduling with sequence-dependent setup times. *Computers & Operations Research*, 90, 142-160.
- Slotnick, S. A. (2011). Order acceptance and scheduling: A taxonomy and review. *European Journal of Operational Research*, 212(1), 1-11.
- Tang, D., Dai, M., Salido, M. A., & Giret, A. (2016). Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Computers in Industry*, 81, 82-95.
- Thevenin, S., Zufferey, N., & Widmer, M. (2016). Order acceptance and scheduling with earliness and tardiness penalties. *Journal of Heuristics*, 22(6), 849-890.
- Vroblefski, M., Ramesh, R., & Zionts, S. (2000). Efficient lot-sizing under a differential transportation cost structure for serially distributed warehouses. *European Journal of Operational Research*, 127(3), 574-593.
- Wan, L., & Zhang, A. (2014). Coordinated scheduling on parallel machines with batch delivery. *International Journal of Production Economics*, 150, 199-203.
- Wang, X., Xie, X., & Cheng, T. (2013). Order acceptance and scheduling in a two-machine flowshop. *International Journal of Production Economics*, 141(1), 366-376.
- Xie, X., & Wang, X. (2016). An enhanced ABC algorithm for single machine order acceptance and scheduling with class setups. *Applied Soft Computing*, 44, 255-266.
- Yin, Y., Cheng, T., Wu, C.-C., & Cheng, S.-R. (2014). Single-machine batch delivery scheduling and common due-date assignment with a rate-modifying activity. *International Journal of Production Research*, 52(19), 5583-5596.
- Yin, Y., Wang, Y., Cheng, T., Wang, D.-J., & Wu, C.-C. (2016). Two-agent single-machine scheduling to minimize the batch delivery cost. *Computers & Industrial Engineering*, 92, 16-30.
- Zhang, B., Pan, Q.-k., Gao, L., Zhang, X.-l., Sang, H.-y., & Li, J.-q. (2017). An effective modified migrating birds optimization for hybrid flowshop scheduling problem with lot streaming. *Applied Soft Computing*, 52, 14-27.



© 2019 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).