Integrated scheduling of production and rail transportation

M. Hajiaghaei-Keshteli, M. Aminnayeri⁎, S.M.T. Fatemi Ghomi

Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, 424 Hafez Avenue, Tehran, Iran

A R T I C L E   I N F O

Article history:
Received 9 September 2013
Received in revised form 14 February 2014
Accepted 24 May 2014
Available online 9 June 2014

Abstract

Nowadays, the most used, efficient and inexpensive mode of transportation between supply chain partners is the rail transportation. To the best of our knowledge, there is no research reported to date, which addresses the problem of synchronization of production and rail transportation. In this work, an integrated production and transportation model, which considers rail transportation, is firstly developed to deliver the orders from a facility to the customers (warehouses). The problem is to determine both production schedule and rail transportation allocation of orders to optimize customer service at minimum total cost. Different destinations of the trains, trains' capacities, and different transportation costs are the main aspects of the work which are considered. In order to tackle such an NP-hard problem, a heuristic, two metaheuristics and some related procedures are developed. Besides, Taguchi experimental design method is utilized to set and estimate the proper values of the algorithms' parameters to improve their performance. For the purpose of performance evaluation of the proposed algorithms, various problem sizes are employed and the computational results of the algorithms are compared with each other. Finally, we investigate the impacts of the rise in the problem size on the performance of our algorithms.

Keywords:
Supply Chain Optimization
Integrated production–distribution
Rail transportation
Single machine scheduling
Taguchi experimental design

1. Introduction

Production is a key function within a supply chain. Traditional scheduling models just focused on the determination of schedules for production such that some performance measures are optimized. The coordination between the production schedule and the delivery plan had not been considered in those models. They implicitly assume that there are ample resources available for delivering finished works to their destinations without any delay. However, in reality, the number of conveyances is limited and they may need to serve more than one customer location to increase their utilization.

Recently, the coordination of activities along different partners of a supply chain has received a lot of attention in production and operations management. Rather than considering production scheduling in isolation, several researchers in this decade have designed models that integrate several of the operational functions to optimize customer service at minimum total cost (Potts & Strusevich, 2009). Rearrangement or interruption of production schedules, idle time, tardiness penalties, and unnecessarily high inventories often drive up costs considerably (Ullrich, 2013). In order to reduce total logistics cost, production and delivery operations need to be well coordinated, first within single companies and then within whole supply chains (Manoj, Gupta, Gupta, & Sriskandarajah, 2008). One of the most interesting issues in supply chain management is the coordination of production and delivery schedules. In many production systems, finished products are delivered from a factory to multiple distribution centers, warehouses, or customer locations by delivery conveyances. The decision maker wants to minimize the total production and distribution costs of a product regarding to customer's satisfaction. So, integrated production and distribution planning have received a lot of attention throughout the recent years and its economic advantages are well documented (Amorim, Gunther, & Almas-Lobo, 2012). Besides, it has been the subject of significant debates, both in academia and industries. The main reasons of the integration are increasing levels of global competition, creating a more demanding customer, demand driven markets and the emergence of just in time delivery.

Generally, researches in this area are mainly focused on road transportation and there has been a relative neglect of other transportation modes while there are sizable sectors of industry such as air and rail transportation. Here, because of the importance of rail transportation in distribution systems, we aim to study the problem of integration of production and rail transportation scheduling to achieve accurate scheduling minimizing total cost. The main difference between rail or air transportation with road transportation is the fixed departure time of the formers. Also the difference between rail and air transportation is that, in the rail transportation a train can transfer the orders of multiple customers which
are in the same pass in the railway (like a tree in a graph). In freight rail transportation, orders and trains may have different destinations, so the orders should only be allocated to the trains with the same destinations or pass through the orders' destinations. So, we should ensure that by allocating an order to a destination with a train in the railway, we do not neglect the farther destinations' orders. Because this type of coordination is firstly discussed in the literature, we explain the reasons which motivate us to work on this subject in Section 3.

We study an integrated production and delivery scheduling problem for a facility faced by a make-to-order company and utilized rail transportation to deliver its orders to the customers (warehouses). At the beginning of a planning horizon, the company has accepted a set of orders and committed a delivery date for each order. The company needs to process these orders on a dedicated production line and deliver the finished orders to the respective customers by the pre-planned trains with their corresponding capacities and traveling times. The problem is to determine a production schedule for the accepted orders and to allocate the orders to the available trains for delivering each completed order subject to the constraint that all the orders are completed and delivered to their customers on the respective committed delivery dates. In this system, earliness and tardiness penalties happen at both production site and at the warehouses.

The ‘departure time earliness’ cost happens as a result of the need to store the order at the production facility or waiting charges at the railway station. Also, there is an extra cost corresponding to charter trains considered as ‘departure time tardiness’ when shipping in a scheduled train is missed or there is no scheduled train for a specific time interval. Delivery penalties are incurred by delivering an order either earlier or later than the committed due date to customers. The ‘delivery tardiness’ cost includes customer dissatisfaction, contract penalties, loss of sales, and potential loss of reputation for manufacturer and retailers. If the arrival time of allocated orders in rail transportation model is earlier than its due date, retailers encounter the storing cost of orders which is considered as 'delivery earliness' cost. In this paper, we schedule both the production and allocate the trains in a way to minimize the total cost.

Lives related works, which presented new approaches and models about coordination in supply chain, e.g. Rostamian Delavar, Hajiaghaei-Keshhteli, and Molla-Alizadeh-Zavaredehi (2010), Li, Ganesan, and Sivakumar (2005), Li, Vairaktarakis, and Lee (2005), Li, Ganesan, and Sivakumar (2006), Li, Sivakumar, and Ganesan (2008), we utilized metaheuristics and heuristics which are relevant to the nature of the problem, too. We utilize the GA with SA which are different in the search approach. The GA is a population-based algorithm where the SA is not. While SA creates a new solution by modifying only one solution with a local move, GA also creates solutions by combining two different solutions. Comparing these two algorithms, in fact, is to consider two different search methods in metaheuristics.

The remainder of the paper is organized as follows. The next section is devoted to the literature review of related topics. In Section 3, the relation between industry and rail freight transportation is discussed. In Sections 4 the assumptions are detailed and the mathematical model is formulated. In Sections 5 and 6, the proposed algorithms are explained and also the Taguchi experimental design and comparison of the computational results are detailed in Section 7. Finally, in Section 8, the paper is concluded and suggestions for the future research are proposed.

2. Literature review

In the supply chain literature, most researchers have focused on the production–distribution, from strategic, tactical, or operational perspectives. Research on integrated scheduling models of production and distribution is relatively recent, but is growing very rapidly. In many applications including make-to-order or time-sensitive (e.g., perishable, seasonal) products, finished orders are often delivered to customers immediately or shortly after the production. Therefore, the need for integration between supply chain partners is inevitable for these products.

Supply chain models that focus on integrating production and distribution operations (that is, on manufacturer–customer levels of the supply chain) are known in the recent literature under the name of integrated production–distribution problems (Chen & Vairaktarakis, 2005; Pundoor & Chen, 2005; Russell, Chiang, & Zepeda, 2008). The authors refer the readers to see the related papers in integrated analysis of production–distribution systems (Armentano, Shiguemoto, & Lekketangen, 2011; Ashoka Varthanana, Muruganb, & Mohan Kumarc, 2012; Chen, 2004, 2010; Chen & Vairaktarakis, 2005; De Matta & Miller, 2004; Erenguc, Simpson, & Vakharia, 1999; Goetschalckx, Vidal, & Dogan, 2002; Melo & Welsey, 2010; Sarmiento & Nagi, 1999). In most production–distribution systems, finished jobs are delivered from a facility to the warehouses (customers) by vehicles such as trucks. There have been considerable researches conducted in production–transportation integration with emphasis on the road transportation and vehicle routing problem (Chang & Lee, 2004; Chen & Lee, 2008; Lee & Chen, 2001; Tang & Liu, 2009; Wang & Cheng, 2006; Xuan, 2011; Zhong, Dosa, & Tan, 2007).

The literature on production–distribution is vast ranging in multiple parameters. According to an outstanding review in this research area written by Chen (2010), there are five delivery methods considered in the previous papers as production–distribution works. The last delivery modulus which is mentioned in the paper, is a delivery method with fixed departure dates. This way gains lots of attention in today's world, but few papers have been published about it. Over 70% of the companies worldwide now rely on third party logistics (3PL) providers for their daily distribution and other logistics needs (Langley, van Dort, & Sykes, 2006). The statistics about using the 3PL shows the rapid growing of using this method of delivery in real world, but only about ten papers have been written about the integration of production and this method of delivery. To the best of our knowledge, the transporters which are considered in the previous papers for fixed departure dates are cars and airplanes.

There is a few research on production scheduling considering vehicles with fixed departure dates. Most of them use air transportation. Li, Sivakumar, Mathirajan, and Ganesan (2004) studied the production of single machine scheduling and air transportation with single destination. The overall problem is decomposed into air transportation problem and single machine scheduling problem. They formulated two problems and then presented a backward heuristic algorithm for single machine scheduling. They extended their previous work to consider multiple destinations in air transportation problem (Li, Ganesan, et al., 2005; Li, Vairaktarakis, et al., 2005). They also proposed a forward heuristic and a backward heuristic for single machine (Li et al., 2006). Li et al. (2008) extended their work by considering parallel machines in production and using simulated annealing (SA) based heuristic algorithm to solve the problem. Zandieh and Molla-Alizadeh-Zavaredehi (2008) also proposed some mathematical models for both production and transportation problems with different delivery assumptions (with delivery tardiness and without delivery tardiness) regarding due date. Zandieh and Molla-Alizadeh-Zavaredehi (2009) extended their work by considering various capacities with different transportation cost and also charter flights (commercial flights). Besides, Rostamian Delvar et al. (2010) proposed two genetic algorithm (GA) approaches to determine both production schedule and air transportation.
allocation of orders to optimize customer service at minimum total cost.

Application of heuristics and metaheuristics to handle these integrated problems is also one of the issues that can be highlighted and discussed in the literature review. Until now, because of the nature of the problems, there are lots of heuristics and metaheuristics have been utilized. The authors refer the readers to see some of the recent works like, Rostamian Delavar et al. (2010), Li et al. (2008), Armentano et al. (2011), Condotta, Knust, Meier, and Shakhlevich (2013), Calvete, Gale, and Oliveros (2011), and Russell et al. (2008) which used metaheuristics like GA, SA, Ant Colony Optimization, and Tabu Search and also Li et al. (2004, 2006) which developed several heuristics for their problems. Besides, a recent outstanding review on integrated production–distribution, written by Fahimnia, Zanjirani Farahani, Marian, and Luong (2013), classified the previous works in this research area by some criteria such as models, complexity, and soft computing techniques.

So, from the literature, it can be concluded that this research area is an important topic both in academia and industry. Besides, to the best of our knowledge, there is no research focusing on an integrated scheduling of production and rail transportation in terms of production–distribution coordination in supply chain.

3. Industry and rail transportation

In this section, in a nutshell, we review the relation between industry and rail transportation. We mention the main aspects of this relation and reference the general advantages to show the importance of this relation and cooperation.

Freight transportation is the process of conveying different types of goods from one point to another by using a variety of transportation modes. The transport of freight can involve road solutions, air and rail deliveries and even the use of waterways to move the freight from a point of origin to one or multiple destinations.

Freight transportation plays a vital role in the economies of nations, regions, and industries (Forkenbrock, 2001). Since 30% of an item price is incurred in the distribution process, lots of companies are currently trying to develop new distribution strategies to efficiently manage their product flow (Apte & Viswanathan, 2000).

Railway transport occupies a significant role in the transportation system of a developed or developing country. The freight rail transportation has some advantages which make it distinct from other transportation modes. It facilitates long distance transport of bulky goods which are not easily transported through motor vehicles. The ability of trains to haul large quantities of goods over long distances is the mode’s primary asset. It also helps in the industrialization process of a country by easy transportation of coal and raw-materials at a cheaper rate with higher speed and certainty. Moreover, the traffic can be protected from the exposure to sun, rain, snow etc. It is a reliable mode of transportation at the time of emergencies like famines and scarcity, too. Another main advantage is the carrying capacity of the railways which is enough large to transfer the big goods. Moreover, in most countries, rail infrastructures are embedded in ports, terminals, boarders, airports, mines and industrial parks to other places and regions with the variety of wagons.

According to the Association of American Railroads (ARR), over the last three decades, rail transportation costs remain approximately flat while consumer goods become pricier. Because transportation cost has an important role in the final price of product, the low cost of rail transportation is a competitive advantage in today’s market and supply chain.

Freight rail also delivers green transportation. Moving freight by rail reduces greenhouse gas emission. Fig. 1 shows the role of rail transportation in producing CO₂ emissions. Use of freight rail offers a simple, inexpensive, safe, and immediate way to reduce greenhouse gas emissions without harming the economy. On average, railroads are four times more fuel efficient than trucks. According to the ARR, if just 10% of long-haul freight now moving by truck moved by rail instead, annual greenhouse gas emissions would fall by more than 12 million tons, saving a cumulative total of nearly 200 million tons by 2020.

So, due to above explained reasons, the rail freight transportation plays a vital role in goods transportation.
As depicted in Fig. 2, it may be concluded that the development of trade, industry, and commerce of a country depends on the development of railways.

The cooperation between industries and rail transportation in order to move their goods or raw materials is very close in most of industrialized countries. Many companies need to reduce their costs to survive in highly competitive global markets. Furthermore, they have a great desire to satisfy their customer’s needs within safe, low-cost, fast and in time delivery. This issue motivates us to work on integration of the production schedule and rail freight transportation in this paper.

4. The problem formulation and descriptions

Here, we firstly develop a mathematical model for the integrated production and rail transportation scheduling problem. We want to determine an optimal allocation of orders to available transportation capacities and also specify sequence and completion times of these orders in production in such a way that the total cost of supply is minimized. In this system, earliness and tardiness penalties happen at both production site and at the warehouses.

The departure time earliness and tardiness costs happen as a result of the need to store the order at the production facility or waiting charges at the rail station, and when shipping in a scheduled train is missed or there is no scheduled train for a specific time interval, respectively. Also the ‘delivery tardiness’ cost includes customer dissatisfaction, contract penalties, loss of sales, and potential loss of reputation for manufacturer and retailers and delivery earliness cost happens when the arrival time of allocated orders in rail transportation model is earlier than its due date.

The assumptions used in this problem are:

1. The production site is treated as a single machine.
2. No idle time is allowed.
3. There are multiple trains in the planning period with different transportation specifications such as time, capacity, etc.
4. Local transportation transfers products from the plant to the rail station. Local transportation time is assumed to be included in transportation time.

5. Business processing time and cost, together with loading time and loading cost for each train are included in the transportation time and transportation cost.
6. Local transportation can transfer an order to the rail station when the order is produced completely.
7. There are no production backlogs, meaning that the orders released into production facility for the planning period are delivered within the same planning period.

The notations that will be used to describe the problem and in the rest of this paper, are as follows:

<table>
<thead>
<tr>
<th>Indices</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i, \bar{i})</td>
<td>order/job index, (i, \bar{i} = 1, 2, \ldots, N)</td>
</tr>
<tr>
<td>(t)</td>
<td>ordinary train index, (t = 1, 2, \ldots, T)</td>
</tr>
<tr>
<td>(k)</td>
<td>order destination index, (k = 1, 2, \ldots, K)</td>
</tr>
<tr>
<td>(l)</td>
<td>pass (train destination) index, (l = 1, 2, \ldots, L)</td>
</tr>
<tr>
<td>(j, \bar{j})</td>
<td>position or sequence of order (ij, \bar{j} = 1, \ldots, N)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_t)</td>
<td>departure time of ordinary train (t) at the local rail station; without loss of generality, it is assumed that (D_1 \leq D_2 \leq \ldots \leq D_T)</td>
</tr>
<tr>
<td>(A_{it})</td>
<td>arrival time of ordinary train (t) at order (i)’s destination;</td>
</tr>
<tr>
<td>(Q_i)</td>
<td>quantity of order (i);</td>
</tr>
<tr>
<td>(x_{it})</td>
<td>departure time earliness penalty cost (/unit/h) of order (i);</td>
</tr>
<tr>
<td>(y_{it})</td>
<td>delivery earliness penalty cost (/unit/h) of order (i);</td>
</tr>
<tr>
<td>(\beta_{it})</td>
<td>delivery tardiness penalty cost (/unit/h) of order (i);</td>
</tr>
<tr>
<td>(d_i)</td>
<td>due date of order (i);</td>
</tr>
<tr>
<td>(Cap_t)</td>
<td>capacity of train (t);</td>
</tr>
<tr>
<td>(T_{c1i})</td>
<td>transportation cost of each product unit when allocated to ordinary train (t);</td>
</tr>
<tr>
<td>(T_{c2i})</td>
<td>transportation cost of each product unit when allocated to charter train (t);</td>
</tr>
<tr>
<td>(\omega)</td>
<td>a real positive digit;</td>
</tr>
<tr>
<td>(Des_{it})</td>
<td>0 if order (i) can be transferred by train (t, \bar{z} \neq 0) otherwise;</td>
</tr>
<tr>
<td>(des_t)</td>
<td>destination of train (t): the train’s final destination</td>
</tr>
</tbody>
</table>

(continued on next page)
With respect to the problem defined above and mentioned assumptions, the mathematical model is formulated as follows:

\[
\begin{align*}
\min & \sum_{i=1}^{N} \left( \left( \min \left( 0, c_i - D_i - \frac{b_i}{t} \right) \right) \times (T_{ci} \times x_t) + (x_t \times (D_i - c_i) \times x_u) \right) \\
& + \left( (x_t \times \max(0, d_i - A_i) \times x_t) + (b_i \times \max(0, A_i - d_i) \times x_u) \right) \\
& + \left( 1 - \left( \min \left( 0, c_i - D_i - \frac{b_i}{t} \right) \right) \times (T_{ci} \times x_t) + \left( \min(x_{ti}, x_{ti}) \times (\max(0, MD_i - c_i) \times x_t) \right) \\
& + \left( b_i \times \max(0, c_i - MD_i) \times y_i \right) \right)
\end{align*}
\]

Such that :

\[
\begin{align*}
x_{it} & \times \prod_{i=1}^{t} (des_t - des_{i-1}) = 0 & i = 1, \ldots, N; t = 1, \ldots, T \\
\sum_{i=1}^{N} x_{it} & \leq \text{Cap}_t & t = 1, \ldots, T \\
\sum_{t=1}^{N} x_{it} + y_i & = Q_i & i = 1, \ldots, N \\
\sum_{j=1}^{N} u_{ij} & = 1 & i = 1, \ldots, N \\
\sum_{i=1}^{N} u_{ij} & = 1 & j = 1, \ldots, N \\
\sum_{j=1}^{N} (u_{ij} p_i Q_i + \sum_{j=1}^{N} u_{ij} p_i Q_j) & = c_i & i = 1, \ldots, N \\
u_{ij} & \in \{0, 1\} & i = 1, \ldots, N; j = 1, \ldots, N \\
x_t & = \text{Non-negative integer variable} \\
y_i & = \text{Non-negative integer variable}
\end{align*}
\]

The integrated model considers both transportation allocation and production scheduling. In other words, orders allocation to the ordinary and charter trains, and also determination of sequence and completion time of orders are considered in one model. Some orders might not reach to their allocated trains and must be transported by charter trains. Consequently, the objective function includes three parts. The first part is about the allocated orders which can reach to their ordinary trains and includes transportation cost of ordinary trains, departure time earliness penalties, and total delivery earliness tardiness penalties. Also, the allocated orders which cannot reach to their ordinary trains are considered in the second part of the objective function. Finally, the allocated charter trains are considered in the last part. Since the orders considered in the last two parts should be transported by charter trains, their costs are the same. These two parts involves transportation cost of charter trains, minimum of departure time and delivery earliness penalties, and delivery tardiness penalties. Those orders which are supposed to be transported by charter trains and their completion time is less than their MD, if \( x_{ti} \leq x_{ti} \), departure time of charter trains \( i = MD_j \) and if \( x_{ti} \geq x_{ti} \), the departure time will be \( c_i \). This happens because the departure time of charter trains are arbitrary. So, we consider \( min(x_{ti}, x_{ti}) \) as the earliness penalty. Fig. 3, shows when the types of costs occur in the system.

Constraint sets set (2) ensures that order \( i \) can be allocated to trains which travel to or pass from the order's destination. Constraint sets (3) states that the capacity of ordinary train \( t \) is not exceeded. Constraint sets (4) ensure that order \( i \) is completely allocated. Constraint sets (5) and (6) state that each job has to be assigned to a position, and each position has to be covered by a job. Constraint sets (7) calculate completion time of the jobs in the facility.

4.1. Focusing on the main constraints

Here, we discuss more on the constraint sets (1) and (2) which are special constraint sets used in the rail transportation part.

We can rewrite constraint sets (1) (the objective function) by the following notation:

\[
\begin{align*}
\min & \sum_{i=1}^{N} \sum_{t=1}^{T} \left( T_{ci} \times x_t + (x_t \times (D_i - c_i) \times x_u) \\
& + (x_t \times \max(0, d_i - A_i) \times x_t) + (b_i \times \max(0, A_i - d_i) \times x_u) \right) \\
& + \frac{1}{\text{LN}} \times (\min(x_{ti}, x_{ti})) \times (\max(0, MD_i - c_i) \times x_t) + b_i \times \max(0, c_i - MD_i) \times y_i \right)
\end{align*}
\]

and also adding the following constraint sets to the mathematical model's body:

\[
\begin{align*}
c_i & \leq \left( \min \left( \frac{D_i}{\max(0, x_t - b_i) \times \frac{1}{x_t - b_i}} \right) \right) \left( 1 + \frac{1}{\text{LN}} \right) & i = 1, 2, \ldots, N
\end{align*}
\]

The constraint sets (12) ensure that to allocate an order to ordinary trains, the completion time of the order should be less or equal to minimum departure time of the allocated trains. So, the first and second part of constraint sets (1) are merged and altered to constraint (11).

By the following example we elucidate the constraint sets (12). Suppose that we have just one order, \( Q_1 = 100 \), and it can be allocated to the three trains numbered 2, 4 and 8 out of 10 trains. The departure times of the trains are 7, 9 and 13, respectively, as depicted in the Fig. 4.

Suppose that the order should be transported by all the three trains such that \( x_{12} = 30, x_{14} = 50, x_{18} = 20 \). The processing time of the order is 6 which is equal to the completion time in this example because we have just one order. So, this order can be allocated to the three trains, because \( c_1 = 6 \) is less or equal to \( min(D_2, D_4, D_8) = 7 \).
The constraint sets (12) also can be rewritten to the following constraint (13):

\[
\begin{align*}
&c_i \leq \left( \frac{D_t}{\max(0, x_t - \theta)} + \frac{1}{IN} \right) \left( 1 + \frac{1}{IN} \right) \quad i = 1, 2, \ldots, N, \ t = 1, 2, \ldots, T. \end{align*}
\]

(13)

The constraint sets (13) check all the trains for all orders. Besides, we can use constraint sets (14) or (15) instead of constraint sets (12) or (13).

\[
\begin{align*}
&x_t \times c_i \leq x_t \times D_t \quad i = 1, 2, \ldots, N, \ t = 1, 2, \ldots, T. \quad (14) \\
&x_t \times (c_t - D_t) \leq 0 \quad i = 1, 2, \ldots, N, \ t = 1, 2, \ldots, T. \quad (15)
\end{align*}
\]

To detail more on constraint sets (2), here we study an example shown in Fig. 5. In this example we have a facility, which produces seven orders in a time period for the five warehouses in which the railway connects the facility to the warehouses. There are four trains with different destinations. For example, the train 2 just can transfer orders which should go to warehouse 2 while the train 1 can transfer orders which should go to warehouses 1, 4, or 5. In other words, orders that should be transferred to warehouse 1, can be transported by each of the trains 1, 3, or 4.

For the order 5 which goes to warehouse 1, there are two passes \((L_i = 2)\) that transfer this order to its corresponding destination. So, this order can be transported by the trains which travel on one of these passes. By the following calculation, these trains are 1, 3, and 4.

\[
\begin{align*}
&(des_{s1} - des_{s1}) \times (des_{s2} - des_{s1}) = (3 - 5) \times (5 - 5) = 0 \quad \leftarrow \\
&(des_{s1} - des_{s2}) \times (des_{s2} - des_{s2}) = (3 - 2) \times (5 - 2) = 3 \\
&(des_{s3} - des_{s1}) \times (des_{s2} - des_{s3}) = (3 - 3) \times (5 - 3) = 0 \quad \leftarrow \\
&(des_{s3} - des_{s4}) \times (des_{s4} - des_{s4}) = (3 - 5) \times (5 - 5) = 0 \quad \leftarrow
\end{align*}
\]

Here, we show constraint sets (2) by another notation and concept. We define a parameter \(Des_{x_i}\) to show that the orders can be allocated to the trains which transfer them to their corresponding destination. If an order can be transported to a train, \(Des_{x_i}\) gains zero value and otherwise it gains a nonzero value \((\theta)\) in order to ensure that the order cannot be allocated to a train which does not transfer the order to its corresponding destination. Besides, the following constraint sets (16) can be used instead of constraint sets (2):

\[
\begin{align*}
&x_t \times Des_x = 0 \quad i = 1, \ldots, N, \ t = 1, \ldots, T \quad (16)
\end{align*}
\]

The values of \(Des_x\) for each order and train for the example detailed in Fig. 5 are brought in Table 1.

5. The Proposed GA

The GA, first introduced by Holland (1975), has proven to be particularly useful for solving complex combinational and various types of problems. It is considered to be one of the typical meta-heuristics for tackling both discrete and continuous optimization...
problems. GAs have been shown as a robust optimization technique to solve many real world problems (Gen & Cheng, 2000).

In GA, any potential solution is represented as a chromosome in an initial population. The GA searches a problem space with a population of chromosomes, each of which represents an encoded solution. A fitness value is assigned to each chromosome according to its performance. The more desirable a chromosome, the more the fitness value it acquires. The initial population gets enhanced over the course of several generations. In each generation, new chromosomes are produced by various genetic operators, such as reproduction, crossover and mutation. Chromosomes for the next population are selected from the previous population by a criterion considering their fitness function. This procedure is repeated until a termination condition is met. Some researches in the literature used genetic algorithm in the supply chain environment (Hajaghaei-Keshteli, 2011; Hajaghaei-Keshteli, Molla-Alizadeh-Zavardehi, & Tavakkoli-Moghaddam, 2010; Rostamian Delavar et al., 2010; Yimer & Demirli, 2010; Zegordi, Kamal Abadi, & Beheshti Nia, 2010).

In this work, at first a GA is used to solve the transportation problem and find a solution with minimum transportation cost and then another GA is applied to the achieved solution to generate the production sequence. This process is continued until the stopping criterion met. Stopping criterion is set to be the CPU time (CT) which is dedicated to the entire algorithm. To obtain the stopping criterion for each of the transportation allocation and the production scheduling, we divide the CT in two parts in which a CT percentage is allocated for transportation’s GA and called as CT. The remaining CT considered as PCT for production scheduling.

Other input parameters in the GA are population size of transportation (PST), population size of production (PSP), reproduction percentage of transportation (TPR), reproduction percentage of production (PPR), mutation probability of transportation (TPM), mutation probability of production (PPM). To have a better understanding of the algorithm, it is represented as a flowchart illustrated in Fig. 6.

Since the transportation cost is just a part of the total cost, having a minimum transportation cost does not ensure that the respective solution has the minimum total cost. So, in the proposed algorithm several good production sequences and destinations’ sequences are generated and evaluated in an iterative way in order to reach the optimum solution which has the lower total cost among the generated solutions.

All the parameters and operators used in the proposed GA, are explained in the following subsections.

5.1. Encoding scheme

In freight rail transportation, orders and trains may have different destinations, so the orders should only be allocated to the trains with the same destinations or pass through the orders’ destinations. For instance, in Fig. 7, there are a facility and warehouses which are related via railway, like a tree. In a specific time period, there are ten orders which should be transported to the corresponding destinations (warehouses) with eight pre-planned trains. The destinations of the trains are the final warehouses that a train meets. For example, the destination of the train 3 is the warehouse 6. So, this train can transfer the orders 1, 5, 8, and 9 to their corresponding destinations which are 1, 3, and 6. The orders’ quantities, the trains’ capacities and transportation cost for the example shown in Fig. 7, is depicted in the Fig. 8.

To encode a real solution into a chromosome for using in the proposed GA, we develop a new, capable and easy procedure. To allocate the orders to the trains, at first we should consider the orders which go to the destinations where are the trains’ final destinations. In the other words, the demands of destinations which are at the end of the tree’s branches should be transported by the related trains at first and then in a backward way, other destination’s demands should be satisfied. By this way, we ensure that by allocating an order to a destination with a train in the tree, we do not neglect the farther destinations’ orders. The number of trains going to a destination where locates at the end of a branch, is equal or lower than the ones which go to the internal destinations (not located at the end of the branches). So, when we allocate an order to a train which goes to a warehouse which is not the final destination of the train, there may be no more train to transfer the orders of final destination of that branch. To employ this compelion, we introduce a precedence constraint for allocation of the

| Table 1 |
The values of Desi, for each order and train.

<table>
<thead>
<tr>
<th>Desi</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>λ</td>
<td>λ</td>
<td>0</td>
<td>λ</td>
</tr>
<tr>
<td>2</td>
<td>λ</td>
<td>0</td>
<td>λ</td>
<td>λ</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>λ</td>
<td>λ</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>λ</td>
<td>λ</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>λ</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>λ</td>
<td>λ</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>λ</td>
<td>λ</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 6. The proposed GA. CT: CPU Time, ITGA: Iteration of Transportation’s GA, BDS: Best Destinations Sequence, BPS: Best Production Sequence, TBPS: Best Production Sequence, TBTA: Total Best Transportation Allocation, TBPS: Total Best Production Sequence.
In order to elaborate the upcoming procedure obviously, which limits the allocation sequence of destinations, we bring an example and explain the precedence constraint which must be used in the example shown in Figs. 7 and 8. Table 2 shows the precedence of the location of the warehouses shown in Fig. 7.

Having a precedence table like Table 2, we can design the relevant tree. In the other words, there is a one-to-one mapping between the table and its relevant tree. First row in the Table 2 shows warehouse 1 is located before warehouses 3 and 4. The same logic is true for the row 3 which shows that warehouse 3 precedes warehouses 5 and 6. The warehouses that their corresponding summation or value of $\theta$, are equal to zero, are located in the end of branches in the tree. So, we should satisfy the warehouses’ demands which their values of $\theta$ are equal to zero. By developing the following procedure we explain the precedence constraint in assigning the warehouses’ demands:

5.1.1. Procedure: Using precedence constraint to develop the destinations’ sequences

Input: Precedence table according the tree
Output: Feasible sequence of destinations for assigning demands (sequence array $O$)

Step 1: Compute the value of $\theta$, which is the summation of values 0 and 1 in the relevant row, for each warehouse.

Step 2: Select one destination which its corresponding $\theta$ is equal to zero. If more than one $\theta = 0$ exist, select one of them randomly.

Step 3: Put the number of the selected destination in the leftmost position of array $O$ (Let $O$ be the sequence array of destinations which their demands should be assigned).

Step 4: Delete the row and column of the selected destination from the precedence table.

Step 5: If still one or more than one destinations exist in the precedence table, go to step 1, otherwise stop.

For the example explained in Fig. 7, one of the feasible destinations sequence (chromosome), can be generated according to the procedure, is as follows:

$$O: 5 \ 4 \ 6 \ 2 \ 3 \ 1.$$ 

Having the sequence of destinations for allocation of the trains to transport their demands, leads to have several sub-matrices in the number of the warehouses to allocate. As explained earlier, each order can be allocated to one or more than one trains which go to its destination. The orders explained in the Fig. 7 can be allocated to the trains according to the Fig. 9.

For example, according to the sequence of destinations, at first step, we should allocate the demands of destination 5 to the trains that can transport these demands to this destination according to the Fig. 9. So, the sub-matrix is generated as shown in Fig. 10 and it is ready for allocation in an arbitrary way or one of the ways explained in the following section, like Vogel’s Approximation Method, or etc. After allocating the orders which must transported to destination 5, extra capacity may still remain in the trains, as shown in Fig. 11. Without loss of generality, we supposed that the trains’ capacities are equal or greater than the total orders’ quantity in a period. However, as explained in the mathematical model, using the charter trains is also another alternative.
This way is continued until satisfying demands for all destinations according to the destinations sequence (the chromosome).

For the production scheduling, the order of jobs to be operated on single machine are put in an array and we consider each possible array as a chromosome.

5.2. Initial procedure

The initial solution has a significant role on the final result’s quality of a search procedure. It has already been recognized and emphasized by many researchers in the recent years (M’Hallah, 2007; Rostamian Delavar et al., 2010). What has been utilized so far by the majority of researchers to generate the initial solution for their algorithms has been the random generation of the initial solution, which has led them to poor quality solutions. Therefore, in order to acquire a satisfactory level of solution quality for such a hard combinatorial problem, meticulous considerations should be given to the intelligent selection of their initialization procedure. In this work, several types of initial procedures have been used. In the following subsections we describe the initial procedures which used for each of rail transportation and production scheduling.

5.2.1. Initial procedures used in rail transportation

The most used heuristic methods to initiate and generate new population in transportation literature are Northwest Corner Method, Row Minimum Method, Column Minimum Method, Least-Cost Method, and Vogel’s Approximation Method. Generally, the Vogel’s Approximation Method mostly generates better solution among other methods. One can use one of the mentioned methods to generate the initial solution for their algorithms. After generating the initial solution for each destination in a sub-matrix, the solution should be optimized by the method of multipliers. So, each destinations sequence (chromosome) maps just one allocation. We generate chromosomes up to the size of population.

\[
\text{Fitness Value} = \frac{1}{\text{Objective Function}}
\]

5.2.2. Initial procedures used in production scheduling

Five initial procedures are used here; SPT, LPT, EDD, EMD, and also random generation. In SPT and LPT, a table is constructed to list the jobs which are arranged in shortest and longest processing time order, respectively. In EDD, jobs are sorted on the basis of earliest due date. In EMD, arranging the jobs is based on maximum departure time, and jobs are listed in descending way. Since, an order has a different destination and transportation time, arranging with EMD method is not similar to EDD.

5.3. Selection mechanism

Regarding the detailed mathematical model, we are to minimize the objective function. Roulette-Wheel method gives more chance to the solution with greater fitness value. So, we consider the fitness value as follow:

\[
\text{Fitness Value} = 1 / \text{Objective Function}
\]

Considering inverse objective function as a fitness value, the greater fitness value leads a solution to has higher selection chance.

5.4. GA operators

5.4.1. Reproduction

With higher probability, better parents can generate better offspring, so it would be necessary to transfer the best solutions of each generation to the next one. Therefore chromosomes with higher fitness values are more desirable, so the p% of the chromosomes with the greater fitness values are automatically copied to the next generation (elite strategy).

5.4.2. Crossover

Crossover operates on two chromosomes at a time and generates offspring by combining both chromosomes’ features. The purpose is to generate ‘better’ offspring, i.e. to create better sequences after combining the parents. As we assigned p% of the chromosomes of generation to reproduction, the (1 – p)% remaining chromosomes are generated through crossover operator.

Because our proposed model, considers two sections; Rail transportation and Production scheduling, we specified a crossover operator for each one:

5.4.2.1. Crossover operator used in rail transportation. Because the developed chromosome for rail transportation is firstly introduced in this work and it suffers the precedence constraint, we would develop a new crossover operator for the chromosome. The following procedure explains the developed crossover operator:

\[
\text{Procedure: Crossover Operator}
\]

Input: Two chromosomes as parents

Output: Two feasible chromosomes as offsprings

Step 1: One point is randomly selected to divide both selected parents into two separate parts. If n is the number of genes in a chromosome, there are (n–1) crossover points. One of these points is selected with equal probability.

Step 2: Omit the digits which exist in the first part of parent I from the parent 2 and name the new version of parent I to modified parent I. Do the same for parent II.

Step 3: The digits on the first side of parent I and II are inherited by the offspring I and II, respectively.

Step 4: Put modified parent I into second side of offspring I from left to right order. Similarly, put modified parent II into second side of offspring I from left to right order.

This procedure ensures that generated offspring are obliged to consider precedence constraint and consequently the generated offspring satisfy the constraint.

\[
\text{Fig. 10. The sub-matrix for allocation of orders to trains which go to destination 5.}
\]

\[
\text{Fig. 11. Allocation of orders to trains which go to destination 5, randomly.}
\]
offsprings are feasible. To explain clearly, we bring an example in Fig. 12.

5.4.2.2. Crossover operators used in production scheduling. For the crossover operation at chromosome generated for single machine scheduling, three types of operators are used as follows:

5.4.2.2.1. One-point crossover. In this operator, one point is randomly selected for dividing one parent. If \( n \) is the number of symbols (genes) in a chromosome, there are \((n - 1)\) crossover points. One of these points is selected with equal probability. The jobs on one side (each side is chosen with the same probability) are inherited by the offspring from the parent, and the other jobs are placed in the order they appeared in the other parent (Murata & Ishibuchi, 1994). After changing the roles of parents, the same procedure is applied to produce the second offspring.

5.4.2.2.2. Two-point crossover. This operator was proposed by Murata and Ishibuchi (1994). First, two crossover points are randomly selected. The symbols between randomly selected points are inherited from one parent to the offspring. The other symbols are inherited from the other parent by their order placed in their own parent. A crossover mask is simply a binary string with the same length of a chromosome. Random numbers are generated between randomly selected points.

Fig. 12. Generating two offsprings from two parents according to the procedure.

<table>
<thead>
<tr>
<th>Parent</th>
<th>Mask</th>
<th>Offspring 1</th>
<th>Offspring 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent 1: 8 3 2 5 1 4 7 6</td>
<td>1 0 0 1 1 0 1 1</td>
<td>5 4 3 6 1 7 4 2</td>
<td></td>
</tr>
<tr>
<td>Parent 2: 1 7 5 8 4 6 3 2</td>
<td>1 1 1 1 1 1 1 1</td>
<td>8 7 5 3 4 2 1 6</td>
<td></td>
</tr>
</tbody>
</table>

Step 1: Parent 1
Step 2: Modified Parent 1: 4 1 8 7 2 1 5 4 3 2 1 5 4 3 2 |
Step 3: Modified Parent II: 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 |
Step 4: Offspring 1: 4 5 6 5 6 1 1 1 |
| Offspring 2: 1 7 5 3 2 1 2 6 4 6 |

One-point crossover

<table>
<thead>
<tr>
<th>Parent</th>
<th>Mask</th>
<th>Offspring 1</th>
<th>Offspring 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent 1: 8 3 2 5 1 4 7 6</td>
<td>1 1 1 1 1 1 1 1</td>
<td>5 4 3 6 1 7 4 2</td>
<td></td>
</tr>
<tr>
<td>Parent 2: 1 7 5 8 4 6 3 2</td>
<td>1 1 1 1 1 1 1 1</td>
<td>8 7 5 3 4 2 1 6</td>
<td></td>
</tr>
</tbody>
</table>

Step 1: Parent 1
Step 2: Modified Parent I: 8 3 2 5 1 4 7 6 | 1 1 1 1 1 1 1 1 |
Step 3: Modified Parent II: 1 7 5 8 4 6 3 2 | 1 1 1 1 1 1 1 1 |
Step 4: Offspring 1: 8 3 2 5 1 4 7 6 |
| Offspring 2: 1 7 5 3 2 1 2 6 4 6 |

Two-point crossover

<table>
<thead>
<tr>
<th>Random numbers</th>
<th>Mask: 0 0 0 1 1 1 1 1</th>
</tr>
</thead>
</table>

Fig. 13. Illustration of the operations of three crossover operators.

5.4.2.2.3. Uniform crossover. Uniform crossover initially generates a random crossover mask and then exchanges relative genes between parents in accordance with the mask. The other genes are inherited from the other parent by their order placed in their own parent. A crossover mask is simply a binary string with the same length of a chromosome.

Fig. 13 shows the operations of these three crossovers with eight jobs.

5.4.3. Mutation

The mutation operator is used to rearrange the structure of a chromosome and to slightly change the sequence, i.e. generating a new but similar sequence. The main purpose of applying mutation is to avoid convergence to a local optimum and diversify the population. The mutation operator can also be considered as a simple form of a local search. The probability of mutating an offspring is called the probability of mutation, \( p_m \), which is usually a small number. After generating an offspring by a crossover operator, a random number from uniform [0,1] is dedicated to the offspring. If this random number was less than or equal to \( p_m \), then the mutation operator would be performed on that offspring. As we mentioned earlier about crossover operator, we need to use two types of mutation operators.

5.4.3.1. Mutation operator used in rail transportation. As explained earlier for the crossover operator used in rail transportation, the developed chromosome for rail transportation is introduced in this

Fig. 14. Illustration of mutation operation procedure used in rail transportation.

<table>
<thead>
<tr>
<th>Parent 8 3 2 5 1 4 7 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offspring 8 3 2 5 1 4 7 6</td>
</tr>
</tbody>
</table>

Step 1: Parent: 8 3 2 5 1 4 7 6 |
Step 2: 8 3 2 5 1 4 7 6 |
Step 3: 8 3 2 5 1 4 7 6 |
Step 4: Possible mutated parent: 6 5 4 2 3 1 1 |

Fig. 15. Illustration of four mutation operation.
work for the first time and it suffers the precedence constraint, we would develop a new mutation operator for the chromosome. In the following procedure, the developed mutation operator is explained:

**Procedure: Mutation operator**

Input: A chromosome as parent  
Output: A feasible chromosome as offspring

---

**Step 1:** One gene is randomly selected in the parent chromosome.  
**Step 2:** From the precedence constraint, find that the selected gene (destination) is precedence of which destination(s) in the parent.  
**Step 3:** From the precedence constraint, find that the selected gene (destination) is successor of which destination(s) in the parent.
Step 4: Select one point randomly in the following interval and put the selected gene between to other gene in the interval: right hand side of the most right digit of genes found in step 2 and left hand side of the most left digit of genes found in step 3 in the parent.

This procedure ensures that generated offspring are obliged to consider precedence constraint and consequently the generated offsprings are feasible. To explain clearly, we bring an example in Fig. 14, according to the tree shown in Fig. 7 and Table 2.

5.4.3.2. Mutation operator used in Production Scheduling. We present four mutation operators; Swap Mutation, Big Swap Mutation, Inversion Mutation, and Displacement Mutation. In Swap Mutation, two adjacent genes are swapped while in Big Swap Mutation, two genes are selected and swapped. In Inversion Mutation two positions are selected, and the sub string is inverted between two positions. In Displacement Mutation, a sub string is selected and inserted in a position. The examples of these operators are shown in Fig. 15.

6. The Proposed SA

The SA is a popular local search algorithm (metaheuristic) for solving combinatorial optimization problems (Cerny, 1985; Kirkpatrick, Gelatt, & Vecchi, 1983). SA searches the set of all possible solutions, reducing the chance of getting stuck in local optima by allowing moves to inferior solutions under the control of a randomized scheme. The name of the approach points to a direct analogy with the way that liquids freeze and crystallize or that metals cool and anneal.

The SA algorithm starts with an initial solution and iteratively moves towards other existing solutions. The algorithm generates a neighboring solution $s'$ in the neighborhood of the candidate solution $s$. Then, the change of objective function value, $\Delta s = f(s') - f(s)$, is calculated. In case $\Delta s$ value is smaller than zero, to move to the neighboring solution is acceptable; otherwise, if $\Delta s$ value is greater than zero, to reduce the probability to get trapped in local optima, the SA may accept to move to an inferior neighboring solution depending on a randomized scheme. More precisely, the move is still accepted if $r \leq \exp(-\Delta s/T)$, where $T$ is a control parameter, called temperature, and $r$ is a uniform random number between interval $(0,1)$. SA generally starts from a high temperature and then the temperature is gradually decreased. A search is performed for a certain number of iterations, at every temperature. When the termination condition is met, the algorithm will stop. The SA algorithm for the problem is shown in Fig. 16. The solutions are encoded like explained in the GA, and the parameters are mentioned in the figure. The cooling rate for transportation's SA is $\alpha_{TSA}$ and for the product's SA is $\alpha_{PAS}$. For generating the neighbor solution, both in destinations sequence and production sequence, we use mutation operators used in the GA.

7. Computational experiments

7.1. Instances

In order to evaluate the performance of the existing algorithms developed in this research to solve the problem, a plan is utilized to generate test problems. Table 3 shows the experimental design.

The data required for the problem include the number of jobs, trains and destinations. The number of jobs $N$ ranges from 50 to 200, the number of trains $T$ ranges from 15 to 60, and the number of order and train destinations, $K$ and $L$, range from 10 to 40 and 2 to 5, respectively. The problem size is determined by the number of jobs, the corresponding number of trains, and the number of order and train destinations. The value of $N$ is set equal to 5K and the value of $T$ is set equal to 1.5K for each problem. The destination for each order and each ordinary train is generated from uniform distribution between 1 to the number of destinations of the corresponding problem configuration. Ten different problem sizes are considered for experimental study.

Departure time of each ordinary train is generated from a uniform distribution subject to its destination. The total number of

<table>
<thead>
<tr>
<th>Problem parameters</th>
<th>No. of classes</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of orders ($N$)</td>
<td>10</td>
<td>50, 60, 70, 80, 90, 100, 125, 150, 175, 200</td>
</tr>
<tr>
<td>Number of trains ($T$)</td>
<td>10</td>
<td>15, 18, 21, 24, 27, 30, 37, 45, 52, 60</td>
</tr>
<tr>
<td>Number of order and train destinations ($K$, $L$)</td>
<td>10</td>
<td>$(10, 2)$, $(12, 2)$, $(14, 3)$, $(16, 3)$, $(18, 3)$, $(20, 4)$, $(25, 4)$, $(30, 4)$, $(35, 5)$, $(40, 5)$</td>
</tr>
<tr>
<td>Precedence table</td>
<td>1</td>
<td>Randomly generated</td>
</tr>
<tr>
<td>Transportation time ($t_i$)</td>
<td>1</td>
<td>Uniform [1,20]</td>
</tr>
<tr>
<td>Order quantity ($q_i$)</td>
<td>1</td>
<td>Uniform [1,20]$N_i$</td>
</tr>
<tr>
<td>Order due date ($d_i$)</td>
<td>3</td>
<td>Uniform [1,20]$N_i$</td>
</tr>
<tr>
<td>Order departure time earliness penalty cost ($s_{a1}$)</td>
<td>1</td>
<td>Uniform [0.4,0.6]/$(Q_{ip}$, $q_i$)</td>
</tr>
<tr>
<td>Order delivery earliness penalty cost ($s_{a2}$)</td>
<td>1</td>
<td>Uniform [0.4,0.6]/$(Q_{ip}$, $q_i$)</td>
</tr>
<tr>
<td>Order delivery tardiness penalty cost ($s_{b}$)</td>
<td>1</td>
<td>Uniform [0.4,0.6]/$(Q_{ip}$, $q_i$)</td>
</tr>
<tr>
<td>Order destination ($D_{es}$)</td>
<td>1</td>
<td>Uniform [0.4,0.6]/$(Q_{ip}$, $q_i$)</td>
</tr>
<tr>
<td>Ordinary train destination ($D_{tr}$)</td>
<td>1</td>
<td>Uniform [0.4,0.6]/$(Q_{ip}$, $q_i$)</td>
</tr>
<tr>
<td>Capacity of train ($C_{ap}$)</td>
<td>1</td>
<td>Uniform [0.4,0.6]/$(Q_{ip}$, $q_i$)</td>
</tr>
<tr>
<td>Transportation cost of each product unit when allocated to ordinary train ($T_{Ca}$)</td>
<td>1</td>
<td>Uniform [1000,30000]</td>
</tr>
<tr>
<td>Transportation cost of each product unit when allocated to charter train ($T_{Cg}$)</td>
<td>1</td>
<td>Uniform [1000,30000]</td>
</tr>
<tr>
<td>Ordinary train arrival time ($A_i$)</td>
<td>1</td>
<td>$D_{tr} + t_i$</td>
</tr>
<tr>
<td>$p_i$</td>
<td>1</td>
<td>Uniform [1,3]</td>
</tr>
<tr>
<td>The unit product processing time of order ($p_i$)</td>
<td>3</td>
<td>Uniform [0.7,0.9]/$(Q_{ip}$, $q_i$)</td>
</tr>
</tbody>
</table>
trains that have the same destination is denoted by $TT_k$. The corresponding trains are assigned to an ordinary train number $TN_t$, which starts from 1 to $TT_k$. The planning period is set to 168 ($=7 \times 24$) hours for the nineteen test problem sets. Orders due date is drawn from uniform distribution. The earliest delivery time of an order is the sum of processing time of the order and rail transportation time. Therefore, the range for an order’s due date is between $Q_i \times p_i + t_i$ and max ($Q_i \times p_i + t_i [0.5 \text{ or } 1 \text{ or } 1.5] \times (\sum Q_i p_i)$), where $Q_i \times p_i$ is the order’s processing time and $t_i$ is the train transportation time to its destination. Initially, a number is specified to every unit of each order as processing time, using uniform [1,3].

Due to 168 h planning period, problem’s size and orders quantity producing all of orders might be impossible because total processing time of entire orders must be less than 168 h. Therefore the upcoming method is acquired to modify the initial processing times. Assuming that the plant is able to produce [0.1,0.3] or [0.4,0.6] or [0.7,0.9] times of the total orders quantity in the planning period, a random number is generated in these intervals which indicates the plant’s production capacity to produce orders totally. Then the unit product processing time of order $i$ is calculated as:

$$p_i = p_i' \times (168 \times \text{uniform}[0.1,0.3]/(\sum Q_i p_i'))$$

$$p_i = p_i' \times (168 \times \text{uniform}[0.4,0.6]/(\sum Q_i p_i'))$$

$$p_i = p_i' \times (168 \times \text{uniform}[0.7,0.9]/(\sum Q_i p_i'))$$

### 7.2. Parameter setting

Because the effectiveness of metaheuristic algorithms depends on the correct choice of the parameters, here, we study the behavior of the different parameters of the proposed algorithms. One of

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Factors and their levels.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factors</td>
<td>GA symbols</td>
</tr>
<tr>
<td>Population size of transportation</td>
<td>A –</td>
</tr>
<tr>
<td>Reproduction percentage of transportation</td>
<td>B –</td>
</tr>
<tr>
<td>Mutation probability of transportation</td>
<td>C –</td>
</tr>
<tr>
<td>Population size of production</td>
<td>D –</td>
</tr>
<tr>
<td>Reproduction percentage of production</td>
<td>E –</td>
</tr>
<tr>
<td>Mutation probability of production</td>
<td>F –</td>
</tr>
<tr>
<td>Type of crossover in production</td>
<td>G –</td>
</tr>
<tr>
<td>$T_T$</td>
<td>–</td>
</tr>
<tr>
<td>$T_P$</td>
<td>–</td>
</tr>
<tr>
<td>MIIT</td>
<td>–</td>
</tr>
<tr>
<td>MIIP</td>
<td>–</td>
</tr>
<tr>
<td>$\alpha_T$</td>
<td>–</td>
</tr>
<tr>
<td>$\alpha_P$</td>
<td>–</td>
</tr>
<tr>
<td>Time percentage of transportation</td>
<td>H</td>
</tr>
<tr>
<td>Type of mutation</td>
<td>I</td>
</tr>
</tbody>
</table>
the methods widely used in the most researches is the full factorial design, which tests all possible combinations of factors (Guh, 2010; Hajiaghaei-Keshteli et al., 2010; Yolmeh & Kianfar, 2012). When the number of factors significantly increases, this method does not seem to be effective. As discussed earlier, for the proposed GA there are 90 test problems, eight 3 level factors, and one 5 level factor in our case that each of which should be run three times. Hence, the total number of running the problem in SA is $90 \times 3^{8} \times 5^{1} \times 10$, that is equal to $9841500$. The factors and their levels are illustrated in Table 4.

Japanese quality consultant Genichi Taguchi popularized a cost-efficient approach, known as robust parameter design. Taguchi (1986) assumed that there are two types of factors which operate on a process: control factors and noise factors. Due to impractical and often impossible omission of the noise factors, the Taguchi tends to both minimize the impact of noise and also find the best level of the influential controllable factors on the basis of robustness (Hajiaghaei-Keshteli et al., 2010). Moreover, Taguchi determines the relative importance of each factor with respect to its main impacts on the performance of the algorithm. A transformation of the repetition data to another value which is the measure of variation is developed by Taguchi. The transformation is the signal-to-noise (S/N) ratio, which explains why this type of parameter design is called a robust design. Here, the term ‘signal’ denotes the desirable value (response variable) and ‘noise’ denotes the undesirable value (standard deviation). So the S/N ratio indicates the amount of variation present in the response variable. The aim is to maximize the signal-to-noise ratio. In the Taguchi method, the S/N ratio of the minimization objectives is as such (Hajiaghaei-Keshteli, 2011):

$$S/N\ \text{ratio} = -10 \log_{10} {\text{(objective function)}}^2$$

For the GA, to select the appropriate orthogonal array it is necessary to calculate the total degree of freedom. The proper array should contain a degree of freedom for the total mean, two degrees of freedom for each factor with three levels ($2 \times 3 = 6$) and four degrees of freedom for the only factor with five levels. Thus, the sum of the required degrees of freedom is $1 + 2 + 8 + 4 = 21$. Therefore, the appropriate array must have at least 21 rows. The same logic is true for the SA, and the appropriate array must have at least $1 + 2 + 7 + 4 = 19$ rows.

The selected orthogonal array should be able to accommodate the factor level combinations in the experiment. Considering the degrees of freedom for the both algorithms, $L_{27}$ ($3^{9}, 9^{1}$) is an appropriate array that satisfies these conditions for both algorithms. For the GA as there are eight 3-level factors in our work, according to Taguchi experimental design procedure (Hajiaghaei-Keshteli et al., 2010), we can keep one column empty. Additionally, since there is a factor with five levels and this scheme offers a factor with nine levels, we should adjust this array to the problem by means of adjustment techniques. Using the dummy level technique we convert a nine-level column into a five-level column. To assign the five-level factor to the nine-level column from the orthogonal array $L_{27}$ ($3^{9}, 9^{1}$), some of these levels are required to be replicated twice. In this research, second, third, fourth and fifth levels are chosen to be replicated twice. It is essential to notice that, after applying these techniques, the obtained array remains orthogonal. Furthermore, the accuracy of these levels that are replicated twice is twice the accuracy of the other level. Table 5 shows the modified orthogonal array $L_{27}$ ($3^{9}, 9^{1}$), where control factors are assigned to the columns of the orthogonal array and the corresponding integers in these columns indicate the actual levels of these factors.

In SA, we have seven 3 level factors, i.e. the GA has one 3 level factor more than the SA. So the orthogonal array $L_{27}$ ($3^{9}, 9^{1}$) should be modified in the same way. Table 6 shows the modified $L_{27}$ ($3^{9}, 9^{1}$) for the SA.

Nineteen test problems, with different sizes, are solved to evaluate the performance of the presented algorithm. The experiments on the GA and SA were based on the $L_{27}$ orthogonal array, therefore 27 different combinations of control factors were considered. Due

### Table 5

<table>
<thead>
<tr>
<th>Trial</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

### Table 6

<table>
<thead>
<tr>
<th>Trial</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Fig. 17. Mean S/N ratio plot for each level of the factors in GA.

Fig. 18. Mean RPD plot for each level of the factors in GA.

Fig. 19. Mean S/N ratio plot for each level of the factors in SA.

Fig. 20. Mean RPD plot for each level of the factors in SA.

Table 7
Average relative percentage deviation (RPD) for the algorithms.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>50j15t</th>
<th>60j18t</th>
<th>70j21t</th>
<th>80j24t</th>
<th>90j27t</th>
<th>100j30t</th>
<th>125j37t</th>
<th>150j45t</th>
<th>175j52t</th>
<th>200j60t</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>6.96</td>
<td>6.88</td>
<td>6.09</td>
<td>7.53</td>
<td>6.53</td>
<td>8.52</td>
<td>6.54</td>
<td>7.56</td>
<td>8.14</td>
<td>7.88</td>
<td>7.262</td>
</tr>
</tbody>
</table>
to stochastic nature of metaheuristics, ten replications were performed for each trial to achieve the more reliable results. Because the scale of objective functions in each instance is different, they could not be used directly. To solve this problem, the relative percentage deviation (RPD) is used for each instance.

\[
\text{RPD} = \frac{\text{Alg}_{\text{sol}} - \text{Min}_{\text{sol}}}{\text{Min}_{\text{sol}}} \times 100
\]

where \( \text{Alg}_{\text{sol}} \) and \( \text{Min}_{\text{sol}} \) are the obtained objective value for each replication of trial in a given instance and the obtained best solution respectively. After converting the objective values to RPDs, the mean RPD is calculated for each trial. To do according Taguchi parameter design instructions, these mean RPDs, are transformed to S/N ratios. The S/N ratios of trials are averaged in each level and the value is shown in Fig. 17. As shown in Fig. 17, in GA, best parameters of factors A, B, C, D, E, F, G, H and I are obviously 3, 2, 1, 2, 2, 2, 3 and 1 respectively. With more investigations by using another measurement, the RPD as shown in Fig. 18, we reach to the same result.

Similar to GA, considering Figs. 19 and 20, all the best parameters for SA are defined as 3, 2, 2, 2, 2, 2, 2 and 2 respectively, according to their alphabetical order.

7.3. Experimental results

In order to be fair, searching time is set to be identical for both algorithms which is equal to 1.9 \( \times N \times T \) milliseconds. This criterion is sensitive to both problem sizes, \( N \) and \( T \). Using this stopping criterion, searching time increases according to the rise of either number of jobs or number of trains. We generate five instances for each of the nineteen problem configuration to avoid bias in the results, i.e. totally 450 instances, different from the ones used for calibration. Each instance is solved three times. We use RPD measure to compare the algorithms. Table 7 shows the results of the experiments for each problem size, 450 data per average. The best performance is obtained by GA with the RPD of 7.262.

To verify the statistical validity of the results, we have performed an analysis of variance (ANOVA) to accurately analyze the results. The results demonstrate that there is a clear statistically significant difference between performances of the algorithms. The means plot and LSD intervals (at the 95% confidence level) for two algorithms are shown in Fig. 21. They do not meet each other and there is no overlap.

Besides, in order to evaluate the robustness of the algorithms in different situations, we analyzed the effects of the problem size on the performance of both algorithms. Fig. 22 shows the interaction between the quality of the presented algorithms and the size of problems. As it is obvious, GA exhibits robust performance, meanwhile the problems size increases. It also shows remarkable performance improvements of GA in large size problems versus SA. Due to remarkable difference between two metaheuristics in large size problems, it is taken out of the further experiment. Comparatively speaking, the proposed GA keeps its robustness in a variety of situations as well as significantly outperforming the other algorithm in all the different cases.

8. Conclusion

Utilizing rail transportation to deliver the finished products from a facility to the customers is widely used in different industries and countries. This issue motivates us to develop a new approach in the production—distribution literature in this paper. After developing the mathematical model for the integrated scheduling of production and rail transportation, we described how to tackle such an NP-hard problem by using two metaheuristic and by different approaches. Two different metaheuristics, GA and SA, a heuristic, and some relevant procedures are developed in this work. By this approach, we could consider the tradeoff between costs occurred in both parties. To adjust the parameters of the proposed genetic algorithms, the Taguchi parameter design method was employed. Consequently, we could cut down the original gigantic experiment combinations. The proposed algorithms with tuned parameters are compared together. As a result, GA generates better solutions than SA. Besides, we examined the behaviors of the algorithms within the several problem sizes, and the results have shown that when the problem size increases, GA represents better results and also keeps its robust performance.

As a direction for future studies, it could be interesting to work on realistic cases and also develop effective metaheuristics incorporating advanced features. Since realistic cases are of interest, consideration of other practical assumptions such as types of production lines, production speed, maintenance, considering more railways conditions, or utilizing fuzzy logic in modeling, could be regarded as an impressive research. There are also potentially unlimited opportunities for researchers to investigate and develop new algorithms based on other metaheuristics (e.g. Imperialist Competitive Algorithm, Particle Swarm Algorithm, Variable Neighborhood Search, and etc.) or other optimization techniques.

References


