A branch and price approach to the two-agent integrated production and distribution scheduling

Ali Gharaei\textsuperscript{a,b,}\textsuperscript{*}, Fariborz Jolai\textsuperscript{b}

\textsuperscript{a} School of Industrial Engineering, College of Engineering, Alborz campus, University of Tehran, Tehran, Iran
\textsuperscript{b} School of Industrial Engineering, College of Engineering, University of Tehran, P.O. Box: 11155-4563, Tehran, Iran

\textbf{ARTICLE INFO}

\textbf{Keywords:}
Supply chain management
Branch and price
Scheduling and distribution
Multi-agent
Column generation

\textbf{ABSTRACT}

The integration and coordination of decisions is one of the new approaches in the supply chain environment. The proper orders scheduling and distribution strategy of final products are two key factors in this integration. Here, an integrated production scheduling and distribution problem with routing decisions is discussed in a multi-site supply chain. This problem has been investigated from a multi-agent perspective in which customers’ sets, as agents, want to optimize a specific objective on his own set of orders. A mixed integer linear programming (MILP) formulation is developed for this problem. Due to the complexity, it is not logical to solve the problem in a straightforward way such as commercial solvers. Hence, a branch and price framework is introduced in which a Bees algorithm (BA) is used to construct initial columns. Various computational experiments are conducted to evaluate the efficiency of the proposed approach. The results show that the proposed algorithm is superior to the standalone branch and price algorithm and MILP solver in terms of computation times and gap.

1. Introduction

The integrated supply chain has become one of the most important topics in today’s published papers in the industrial engineering field. Each supply chain has several layers, such as suppliers, manufacturers, distributor and ultimately customers. Whenever any of these members decide individually, they only optimize their own objectives, regardless of the other members’ interests. But if the decisions are taken centrally and integrated by the supply chain management, it will lead to an overall optimization of the supply chain. Proof of the profitability of integrated decisions in comparison to the individual decisions of each member has been represented in various papers such as Chandra and Fisher (1994) and Chen (2010). For instance, Chandra and Fisher (1994) have shown that in an integrated way, supply chain costs decline between 3% and 20% when scheduling and production planning decisions are integrated. In our paper, we focus on the integration of production scheduling and distribution of orders among customers.

Proper distribution of customers’ orders, which are in different geographic locations, is one of the key factors to the success of supply chains. If the distribution of orders incur delays, it can lose the satisfaction of customers in on-time delivery of orders. Batch delivery is a common distribution strategy in which several orders form a batch, and they are sent toward customers by some vehicles. This strategy is introduced by Cheng and Kahlbacher (1993) in which delivery time of a vehicle starts when the processing time of the last assigned order to this batch is completed. Here, the batch delivery problem with routing decisions among customers is discussed.

Multi-agent scheduling is one of the newest issues in the scheduling field which was introduced by Baker and Cole Smith (2003) and it was extended by Agnetis, Mirchandani, Pacciarelli, and Pacifici (2004). In the classic scheduling problem, the sequence of orders is determined based on one or more objective(s) that these objectives are common among all orders. But, in the multi-agent scheduling, there are several agents that each one has a set of orders and their own particular objective. In this case, agents try to achieve the shared resource of the supply chain according to their objective. We generalize the multi-agent idea to an integrated scheduling and distribution problem, because in other supply chain issues, multi-agent concept can affect the decisions, such as several categories of customers, or multiple births for different purposes.

In accordance with Agnetis, Billaut, Gawiejnowicz, Pacciarelli, and Soukhal (2014), a multi-agent scheduling problem is divided into four categories based on the structure. In the \textit{competing agents} case, agents do not have any common orders, and each order belongs to just one agent. In the \textit{interfering agents}, a set of orders is a subset of the other sets. In this paper, two agents, or two sets of orders $A$ and $B$, is considered where $A \not\subseteq B$. In the \textit{multi-criteria} case, which is the same as the classic multi-objective scheduling problem, there is one set of orders and they...
The integrated production scheduling and distribution and multi-agent scheduling was first introduced in the scheduling book of Baker and Cole Smith (2003). Then, Agnetis et al. (2004) provide different cases of multi-agent scheduling and a constrained optimization approach is used to deal with these cases. Cheng, Ng, and Yuan (2006) discuss a multi-agent scheduling on a single machine to minimize the total weighted number of tardy jobs. A linear combination of objective function is considered to this problem and a fully polynomial-time approximation scheme is proposed to solve it. In Agnetis, Paciarelli, and Pacifici (2007), a single machine scheduling problem is discussed where agents have a number of late jobs and total weighted completion as objective functions. A comprehensive overview of the papers published in this area was made by Perez-Gonzalez and Framinan (2014). A clear definition and a unified framework for this type of scheduling problem are presented. Also, the application, complexity and solution approaches of different cases are discussed. Yin, Wang, Cheng, and Wu (2016) propose a polynomial-time algorithm for the two-agent single-machine scheduling. Batch delivery consideration and a constrained optimization approach are considered. Moreover, Pareto approaches have been recently taken into account for this problem, such as in Yin, Wu, Cheng, Wu, and Wu (2015), Lin et al. (2017) and Gharaei and Jolai (2018). In Yin, Chen, Qin, and Wang (2018), a column generation method is used for the unrelated parallel machines scheduling problem. They consider two agents for this problem in which the first agent wants to minimize the total completion time and the second agent wants to minimize number of tardy jobs. A constrained optimization approach is used in this problem. Also, Yin, Yang, Wang, Cheng, and Wu (2018) investigate an integrated scheduling, inventory, and scheduling problem with two competing agents on a single machine. They discuss several cases of this problem with several objective functions for both agents and propose an exact or high-quality papers as well as papers in which column generation methods are used in the scheduling and routing problems.

In the integration decisions approach, Cohen and Lee (1988) carry out a strategic analysis for the integrated production-distribution systems. Some approximate sub-models and a heuristic optimization approach are introduced for this model. A production-distribution supply chain with one supplier and one or more customers is discussed in Pundoor and Chen (2005) and some heuristics are introduced for this problem. Chen and Vairaktarakis (2005) provide an exact algorithm for the integrated scheduling of production and distribution problem, and they show a comparison between this problem and separated decisions of production and distribution operations. Nishi, Konishi, and Ago (2007) propose an augmented Lagrangian decomposition method for the integrated production scheduling and distribution in an aluminum company. In Maravelias and Sung (2009), papers in the context of integrated production planning and scheduling problem are reviewed and challenges and future research directions are presented. Chen (2010) reviews integrated scheduling models of production and outbound distribution. He presents a unified model representation scheme, classifies existing models, and an overview of the optimality properties, computational tractability of published papers in this field. Ulrich (2013) propose a genetic algorithm approach to solve integrated machine scheduling and vehicle routing with time windows with total tardiness as objective function. Recently, Li, Zhou, Leung, and Ma (2016) propose a non-dominated sorting genetic algorithm to the integrated production and distribution scheduling problem in which objectives are distribution cost and the total customer waiting time. Han, Yang, Wang, Cheng, and Yin (2019) provide an integrated approach in the three-site supply chain with scheduling, distribution, and inventory decisions. They discuss this problem with three objectives of the number of tardy jobs, inventory and distribution cost and also they propose a pseudo-polynomial algorithm for this problem. Same as this paper, Wang, Yin, and Cheng (2018) propose a branch and price framework in a parallel machine environment, unlike our problem which is an integrated problem in the supply chain. They have investigated this problem when the initial sequence of jobs is already known.

Multi-agent scheduling was first introduced in the scheduling book of Baker and Cole Smith (2003). Then, Agnetis et al. (2004) provide different cases of multi-agent scheduling and a constrained optimization approach is used to deal with these cases. Cheng, Ng, and Yuan (2006) discuss a multi-agent scheduling on a single machine to minimize the total weighted number of tardy jobs. A linear combination of objective function is considered to this problem and a fully polynomial-time approximation scheme is proposed to solve it. In Agnetis, Paciarelli, and Pacifici (2007), a single machine scheduling problem is discussed where agents have a number of late jobs and total weighted completion as objective functions. A comprehensive overview of the papers published in this area was made by Perez-Gonzalez and Framinan (2014). A clear definition and a unified framework for this type of scheduling problem are presented. Also, the application, complexity and solution approaches of different cases are discussed. Yin, Wang, Cheng, and Wu (2016) propose a polynomial-time algorithm for the two-agent single-machine scheduling. Batch delivery consideration and a constrained optimization approach are considered. Moreover, Pareto approaches have been recently taken into account for this problem, such as in Yin, Wu, Cheng, Wu, and Wu (2015), Lin et al. (2017) and Gharaei and Jolai (2018). In Yin, Chen, Qin, and Wang (2018), a column generation method is used for the unrelated parallel machines scheduling problem. They consider two agents for this problem in which the first agent wants to minimize the total completion time and the second agent wants to minimize number of tardy jobs. A constrained optimization approach is used in this problem. Also, Yin, Yang, Wang, Cheng, and Wu (2018) investigate an integrated scheduling, inventory, and scheduling problem with two competing agents on a single machine. They discuss several cases of this problem with several objective functions for both agents and propose an exact or high-quality papers as well as papers in which column generation methods are used in the scheduling and routing problems.

In the integration decisions approach, Cohen and Lee (1988) carry out a strategic analysis for the integrated production-distribution systems. Some approximate sub-models and a heuristic optimization approach are introduced for this model. A production-distribution supply chain with one supplier and one or more customers is discussed in Pundoor and Chen (2005) and some heuristics are introduced for this problem. Chen and Vairaktarakis (2005) provide an exact algorithm for the integrated scheduling of production and distribution problem, and they show a comparison between this problem and separated decisions of production and distribution operations. Nishi, Konishi, and Ago (2007) propose an augmented Lagrangian decomposition method for the integrated production scheduling and distribution in an aluminum company. In Maravelias and Sung (2009), papers in the context of integrated production planning and scheduling problem are reviewed and challenges and future research directions are presented. Chen (2010) reviews integrated scheduling models of production and outbound distribution. He presents a unified model representation scheme, classifies existing models, and an overview of the optimality properties, computational tractability of published papers in this field. Ulrich (2013) propose a genetic algorithm approach to solve integrated machine scheduling and vehicle routing with time windows with total tardiness as objective function. Recently, Li, Zhou, Leung, and Ma (2016) propose a non-dominated sorting genetic algorithm to the integrated production and distribution scheduling problem in which objectives are distribution cost and the total customer waiting time. Han, Yang, Wang, Cheng, and Yin (2019) provide an integrated approach in the three-site supply chain with scheduling, distribution, and inventory decisions. They discuss this problem with three objectives of the number of tardy jobs, inventory and distribution cost and also they propose a pseudo-polynomial algorithm for this problem. Same as this paper, Wang, Yin, and Cheng (2018) propose a branch and price framework in a parallel machine environment, unlike our problem which is an integrated problem in the supply chain. They have investigated this problem when the initial sequence of jobs is already known.

Multi-agent scheduling was first introduced in the scheduling book of Baker and Cole Smith (2003). Then, Agnetis et al. (2004) provide different cases of multi-agent scheduling and a constrained optimization approach is used to deal with these cases. Cheng, Ng, and Yuan (2006) discuss a multi-agent scheduling on a single machine to minimize the total weighted number of tardy jobs. A linear combination of objective function is considered to this problem and a fully polynomial-time approximation scheme is proposed to solve it. In Agnetis, Paciarelli, and Pacifici (2007), a single machine scheduling problem is discussed where agents have a number of late jobs and total weighted completion as objective functions. A comprehensive overview of the papers published in this area was made by Perez-Gonzalez and Framinan (2014). A clear definition and a unified framework for this type of scheduling problem are presented. Also, the application, complexity and solution approaches of different cases are discussed. Yin, Wang, Cheng, and Wu (2016) propose a polynomial-time algorithm for the two-agent single-machine scheduling. Batch delivery consideration and a constrained optimization approach are considered. Moreover, Pareto approaches have been recently taken into account for this problem, such as in Yin, Wu, Cheng, Wu, and Wu (2015), Lin et al. (2017) and Gharaei and Jolai (2018). In Yin, Chen, Qin, and Wang (2018), a column generation method is used for the unrelated parallel machines scheduling problem. They consider two agents for this problem in which the first agent wants to minimize the total completion time and the second agent wants to minimize number of tardy jobs. A constrained optimization approach is used in this problem. Also, Yin, Yang, Wang, Cheng, and Wu (2018) investigate an integrated scheduling, inventory, and scheduling problem with two competing agents on a single machine. They discuss several cases of this problem with several objective functions for both agents and propose an exact or
heuristic approach to solve each problem.

Column generation method is a kind of decomposition method which is designed to solve large LP and MILP problems, and is suitable for problems that have many variables and do not have a lot of constraints, such as cutting stock problem with numerous patterns, crew scheduling, and vehicle routing problem. The idea of decomposition is taken from the Dantzig-Wolfe decomposition of the maximal multi-commodity network flows by Ford and Fulkerson (1958) and Dantzig and Wolfe (1960). Then this method was used in Gilmore and Gomory (1961) and Gilmore and Gomory (1963) for the cutting stock integer problem. There are various papers in which column generation based methods is used to solve routing and scheduling problems. Hernandez, Feillet, Giroudeau, and Naud (2016) develop two branch and price frameworks for the vehicle routing problem with time windows in which each vehicle can travel several times. Parragh and Cordeau (2017) use a branch and price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. In Zamorano and Stolletz (2017), a branch and price is presented for the multi-period technician routing and scheduling problem. A branch and price and check approach is introduced in Lam and Van Hentenryck (2016), for the vehicle routing problem with location congestion. Kowalczyk and Leus (2017) develop a branch and price method to the parallel machine scheduling with conflicts. A branch and cut and price algorithm is presented for the capacitated general routing problem in Bach, Lysgaard, and Wehik (2016). Ziebuhr and Kopfer (2016) apply a column generation to the integrated operational transportation planning problem with forwarding limitations. A decentralized approach based on column generation method is used for the multi-resource constrained scheduling problem in the multi-party supply chain.

By reviewing the published papers, several research gaps were found. The integrated scheduling and distribution problem has been discussed in various papers, but there are few papers when routing decisions are taken into account and the multi-site environment is considered. There are very few papers in which an exact approach is used as solution approach of integrated scheduling and distribution problem, and in particular, there is no branch and price approach to this problem. Hence, the contribution of this paper is as follows: A multi-agent viewpoint is introduced for the integrated production scheduling and distribution problem with routing decisions. A parallel multi-site supply chain is used to implement this problem. A branch and price framework is presented for this problem in which a Bees algorithm is applied to generate high-quality initial solutions in each iteration.

3. Problem description

Consider a multi-site supply chain whose sites are located in the different geographic locations. Hence, we can consider that the orders scheduling at different sites will be similar to the identical parallel machines scheduling problem. Each customer from different locations places an order to the supply chain management. The supply chain management assigns orders to the different sites based on the distance between customers and sites and also agents’ objectives. Then, orders are processed on each site, and after completion, each order has to be assigned to a vehicle. Each vehicle can carry orders from different customers and distribute orders among customers by determining the appropriate routing. Therefore, this problem includes the scheduling and routing of orders such that the distribution cost and total tardiness of orders are minimized. This problem is investigated from two-agent point of view. The first agent is set of customers who are willing to minimize the total tardiness. The second agent is set of customers with total transportation cost as objective. The objective of the second agent is kept as the main objective, and the objective of the first agent is considered as threshold and brought in the constraints.

This study is examined in a multi-product and single period case and the production policy is based on Make-To-Order. No setup times are considered for each site. The fleet is homogeneous and the capacity of the vehicles is limited. There are sufficient vehicles and they are always available at each site. Each order has different processing time, due date and size. All orders are available at the beginning of planning horizon.

3.1. Notation

The nomenclature used in this article is as follows: Note that two artificial orders 0 and n + 1 with zero processing times and due dates are added to set of orders. Also, two artificial customers 0 and n + 1 with zero distance are added to set of customer, where customer 0 means the manufacturing site and the customer n + 1 means returning to the site.

Indices:

\( i, j \) Index of orders or customers
\( s \) Index of sites
\( k \) Index of vehicles

Parameters:

\( n \) Total number of orders or customers
\( S \) Number of sites
\( M \) A positive big number
\( p^j \) Processing time of order \( j \)
\( q^j \) Due date of order \( j \)
\( t^j \) Size of order \( j \)
\( t_{ij} \) Transportation time between customer \( i \) and \( j \)
\( t_{ij} \) Transportation time between customer \( j \) and site \( s \)
\( r_{ik} \) Transportation cost from customer \( i \) to \( k \)
\( r_{ik} \) Transportation cost between customer \( i \) to site \( s \)
\( FC \) Fixed cost of each transmission
\( Q \) Vehicle capacity
\( e \) Upper bound of total tardiness

Decision variables:

\( v^s_j \) Equals 1 if order \( j \) is assigned to site \( s \), otherwise 0
\( x^s_{ij} \) Equals 1 if order \( j \) is processed immediately after order \( i \) in site \( s \), otherwise 0
\( y^k_{ij} \) Equals 1 if order \( j \) is assigned to vehicle \( k \) in site \( s \), otherwise 0
\( z^k_{ij} \) Equals 1 if order \( j \) is delivered immediately after order \( i \) by vehicle \( k \) in site \( s \), otherwise 0
\( CB_{ij} \) Processing completion time of all orders in vehicle \( k \) in site \( s \)
\( C^j \) Processing completion time of order \( j \)
\( A^j \) Delivery time of order \( j \)
\( T^j \) Tardiness of order \( j \)

3.2. Mathematical model

\[
\begin{align*}
\min & \quad \sum_{s=1}^{S} \sum_{i=1}^{n} \sum_{k=1}^{S} \sum_{j=1}^{n} f_{ij} v^s_j + \sum_{s=1}^{S} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} z^k_{ij} +
\sum_{s=1}^{S} \sum_{i=1}^{n} \sum_{k=1}^{S} \sum_{j=1}^{n} f_{ij} z^k_{ij} + FC^s \sum_{s=1}^{S} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} z^k_{ij} \\
\text{subject to} & \quad \sum_{j=1}^{n} v^s_j = 1 \quad \forall \ j = 1, ..., n \\
\sum_{i=0}^{n} x^s_{ij} = v^s_j & \quad \forall \ j = 1, ..., n + 1, \forall \ s = 1, ..., S \\
\end{align*}
\]
\[\sum_{j=1}^{n+1} x_{ij}^j = v_i^j \quad \forall \ i = 0, \ldots, n, \forall s = 1, \ldots, S \quad (4)\]
\[x_{ij}^j = 0 \quad \forall \ j = 1, \ldots, n, \forall s = 1, \ldots, S \quad (5)\]
\[x_{ij} + x_{ij}^0 \leq 1 \quad \forall \ i, j = 1, \ldots, n, \forall s = 1, \ldots, S \quad (6)
\]
\[\sum_{k=1}^{n} y_{kj}^k = v_{ij}^j \quad \forall \ j = 1, \ldots, n, \forall s = 1, \ldots, S \quad (8)\]
\[\sum_{j=1}^{n} z_{kj}^{ks} = y_{kj}^k \quad \forall \ j = 1, \ldots, n + 1, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (9)\]
\[\sum_{j=1}^{n+1} z_{kj}^{ks} = 0 \quad \forall \ e = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (10)\]
\[z_{ij}^{ks} = 0 \quad \forall \ j = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (11)\]
\[z_{i(1)}^{ks} = 0 \quad \forall \ e = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (12)\]
\[z_{ij}^{ks} + z_{ij}^{ks+1} \leq 1 \quad \forall \ j, i = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (13)\]
\[\sum_{j=0}^{n} z_{ij}^{ks} - \sum_{j=1}^{n+1} z_{ij}^{ks} = 0 \quad \forall \ e = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (14)\]
\[C_j \geq p_j - M(1 - x_{ij}^j) \quad \forall \ j = 1, \ldots, n, \forall s = 1, \ldots, S \quad (15)\]
\[C_j \geq C_i + p_j - M(1 - x_{ij}^j) \quad \forall \ i, j = 1, \ldots, n, \forall s = 1, \ldots, S \quad (16)\]
\[CB_{ik} \geq C_j - M \left(1 - y_{kj}^k\right) \quad \forall \ j = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (17)\]
\[A_j \geq CB_{ik} + t_{ij} - M(1 - z_{ij}^{ks}) \quad \forall \ j = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (18)\]
\[A_j \geq A_i + t_{ij} - M(1 - z_{ij}^{ks}) \quad \forall \ i, j = 1, \ldots, n, \forall k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (19)\]
\[T_j \geq a_j - d_j \quad \forall \ j = 1, \ldots, n \quad (20)\]
\[T_j \geq 0 \quad \forall \ j = 1, \ldots, n \quad (21)\]
\[\sum_{j \in Agent_1} T_j \leq \epsilon \quad (22)\]
\[v_i^j, x_{ij}^j, y_{kj}^k, z_{ij}^{ks} \in \{0, 1\} \quad \forall i = j = k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (23)\]
\[CB_{ik}, C_j, A_j, T_j, W_j \geq 0 \quad \forall j = k = 1, \ldots, n, \forall s = 1, \ldots, S \quad (24)\]

Eq. (1) is the objective function of the second agent, as main objective function which represents the cost of transportation among customers, from site to the first customer, from the last customer to the manufacturing site and the fixed cost of each transportation. Eq. (2) guarantees that each order has been assigned to exactly one manufacturing site. Set of constraints (3)-(7) is related to the sequencing of orders in each site. An order is either the first order or the last order of sequence in a site, otherwise there are two orders before and after this order for processing. Assigning completed orders to a vehicle for distribution operation is shown in Eq. (8). The delivery sequence of orders to customers is represented in the set of constraints (9)-(13). Note that, a vehicle returns to the site after delivery of all orders. Eq. (14) is the entry and exit balance of the customers’ nodes. That means to each customer only one vehicle is entered and one is exited. The vehicle capacity limit is considered in constraint (15). Calculations of processing completion times of orders are shown in constraints (16) and (17). Moreover, the completion time for all orders in a batch that is equal to start time of delivery orders of a vehicle is calculated in (18). Delivery time of each order is also calculated in constraints (19) and (20). The total tardiness of each order is calculated in (21) and (22). The upper bound of total tardiness as objective function for orders of the first agent is shown in (23). Constraints (24) and (25) describe integrality and non-negativity nature of variables, respectively. It’s clear that this formulation is very complex to solve with a typical solver. Both parallel machines scheduling and vehicle routing problems are NP-hard. Hence, an efficient exact algorithm is applied to this problem. A Danzig-Wolfe decomposition is used to decompose problem to a master problem and two sub-problems. The solution method is described in the following section in detail.

4. Solution procedure

As shown in the previous section, the formulation presented in Section 3.2 is very complicated due to a large number of integer variables. It's not efficient to use commercial optimization software to solve this problem, because it requires a lot of time and memory resources to solve it. The problem under study in this paper is an operational problem and needs to be solved more quickly, so a more efficient algorithm is needed to solve it. On the other hand, our problem has a decomposable structure. So here, an exact decomposition method is introduced. First, using the Dantzig-Wolfe decomposition, the mathematical model of the problem is decomposed to a master problem and two sub-problems parallel machine scheduling and vehicle routing. Then, a column generation approach is used to solve LP relaxation of problem. A branching scheme is designed to achieve integer solutions. Also, a Bees algorithm approach is used to generate initial columns in the column generation method.

4.1. Branch and price

Branch and price is a column generation based method. In this method, the original problem is divided into the master problem and sub-problem(s). The primary master problem involves a limited number of complicated variables of the original problem, so it's known as the restricted master problem(RMP). The reason for using a limited number of variables for the Master problem is that a lot of computing time is required to solve the initial master problem contains a large number of variables. On the other hand, since just linear relaxation of the master problem is solved by a linear method, in each iteration of the simplex method, there are many variables that are not present in the base and their values are zero. Considering these issues, in the column generation method, instead of considering all these columns or variables, only a subset of columns are considered in each iteration.

In this method, at first, the RMP formulation is initialized with a limited number of variables by an innovative method, which in this study a metaheuristic method is used for this part. Then the linear relaxation of RMP is obtained by dropping integer variables. The dual of RMP has to be solved and the obtained dual variables values are should be embedded to the sub-problems. Now, pricing problems(sub-problems) have to be solved and if there is one or more variable(s) or column(s) with a negative reduced cost, the columns with the most negative reduced cost are added to RMP. If there is no column with negative reduced cost and all the decision variables values are an integer, then the column generation algorithm stops, and the optimal solution is obtained. If there is any non-integer variable in the solution, the branching procedure begins. These steps are repeated until there is no any column with a negative reduced cost, and all variables have
integer values and then the column generation algorithm stops. In fact, in the branch and price approach, in each node of the branch and bound tree, a column generation algorithm must be executed to solve linear programming relaxation of the problem. The framework of proposed Branch and Price algorithm is shown in Fig. 2. In the following, each part of the algorithm will be explained by detail.

4.2. Master problem

As mentioned before, due to the complexity of the original problem, this problem is divided into master problem and sub-problem(s). In this section, a set-partitioning type formulation of the master problem is presented. The notation of this problem is introduced here.

Set or indices:

- \( I \) Set of orders or customers
- \( \Psi \) Set of feasible partial schedules of single machine scheduling problem(\( e \in \Psi \))
- \( \Omega \) Set of feasible routes for a vehicle(\( d \in \Omega \))

Parameters:

- \( n \) Total number of orders or customers
- \( S \) Number of sites
- \( d_j \) Due date of order \( j \)
- \( FV_d \) Transportation cost of partial route \( d \)
- \( a^e_j \) Equals 1 if order \( j \) has covered in partial schedule \( e \)
- \( b^e_j \) Equals 1 if order \( j \) has covered in route \( d \)
- \( c^e_j \) Processing completion time of order \( j \) in partial schedule \( e \)
- \( r^e_d \) Processing completion time of all orders in route \( d \)
- \( a^e_jr^e_d \) Delivery time of order \( j \) in route \( d \)
- \( \epsilon \) Upper bound of total tardiness

Decision variables:

- \( x_e \) Equals 1 if partial schedule \( e \) has existed in the optimal solution
- \( z_d \) Equals 1 if partial route \( d \) has existed in the optimal solution
- \( t^e_j \) Tardiness of \( j \) in partial route \( e \)

Mathematical model

\[
\begin{align*}
\min \sum_{d \in \Omega} z_d \left( FV_d + FC \right) \\
\sum_{e \in \Psi} a^e_j x_e = 1 & \quad \forall j \in I \\
\sum_{d \in \Omega} b^e_j z_d = 1 & \quad \forall j \in I \\
\sum_{e \in \Psi} x_e & \leq S \\
\sum_{d \in \Omega} z_d & \leq n \\
\sum_{e \in \Psi} c^e_j a^e_j r^e_d & \leq \sum_{d \in \Omega} r^e_d b^e_j z_d & \quad \forall j \in I \\
x_e, z_d & \in [0, 1] & \forall e \in \Psi, d \in \Omega
\end{align*}
\] (26)

(27)

(28)

(29)

(30)

(31)

(32)

Eq. (26) represent total transportation cost as objective function which is related to the routing problem. Eq. (27) ensures that each order is processed only at one site, and Eq. (28) also ensures that each order is dedicated to only one tour. Constraint (29) ensures that at most \( S \) manufacturing sites are available for a feasible schedule. Also, constraint (30) imposes the total number of available routes for a feasible route. In constraint (31), it is guaranteed that any partial route \( d \) including order \( j \) does not start until the processing time of this order is completed. Binary requirements is described in (32).

Formulation of the master problem has a large number of variables and to use column generation method, it is necessary to solve this problem. On the other hand, due to binary nature of the variables, this column generation approach must be inserted in a branch and bound framework. Hence, in each iteration of the algorithm, the dual version of RMP problem is solved and the values of its dual variables are obtained. These values are embedded in the objective function of pricing problem. By solving the pricing problem, we are looking for a column that has negative reduced cost, and if such column is not found, the column generation algorithm stops. The RMP formulation is the same as master problem formulation with two differences: (1) Instead of sets \( \Psi \) and \( \Omega \), two subsets \( \Psi' \subseteq \Psi \) and \( \Omega' \subseteq \Omega \) are used. (2) Integrality constraint \( x_e, z_d \in [0, 1] \) has been replaced by non-negativity constraint \( x_e, z_d \geq 0 \).

4.3. Pricing problems

To find better solutions than the current solution of the master problem, it is necessary to solve another problem which is called sub-problems or pricing problem. Solving this problem determines whether the column generation method continues or stops in this iteration. To do this in each iteration, the values of the dual variables of the restricted master problem are entered to the sub-problems as inputs. Now,
if the reduced cost of a column is negative, this column enters to the RMP in the next iteration and the column generation algorithm continues. But if the algorithm could not found any column with negative reduced cost, the branching procedure begins.

In this section, two sub-problem are introduced. A single-machine scheduling problem and a single-vehicle routing problem. But before that, according to the constraints of the master model, we introduce the dual variables. Variables $\pi_1, \pi_2, \pi_3, \pi_4$ and $\pi_k$ are defined as dual cost of a column in a single-machine scheduling problem, corresponds to variable $x$, is: $-\left( \sum_{j=1}^{n} \pi_1 a_j^+ + \pi_3 + \sum_{j=1}^{n} \pi_2 a_j^+ d_j^+ \right)$, where $a_j^+$ and $d_j^+$ are the decision variables. Therefore, the pricing problem of single machine scheduling is as follows:

$$\min_{c \in Q} \left\{ \left( \sum_{j=1}^{n} \pi_1 a_j^+ + \pi_3 + \sum_{j=1}^{n} \pi_2 a_j^+ d_j^+ \right) \right\}$$

(33)

Note that there is not any term of single machine scheduling criteria in the objective function of the original formulation. So, this problem is corresponds to single machine scheduling problem with total weighted completion time as objective. In fact, the objective of this problem is to find a sequence of orders that minimize the objective of scheduling sub-problem or reduced cost.

The second sub-problem is related to vehicle routing problem. Reduced cost of a column in a single-vehicle routing problem, corresponds to variable $y$, is: $\left( F_{V_d} + FC \right) - \left( \sum_{j=1}^{n} \pi_1 b_j^+ + \pi_4 + \sum_{j=1}^{n} \pi_k b_j^+ r_d \right)$, where $F_{V_d}, b_j^+$ and $r_d$ are the decision variables. Therefore, the pricing problem of vehicle routing is as follows:

$$\min_{c \in Q} \left\{ \left( F_{V_d} + FC \right) - \left( \sum_{j=1}^{n} \pi_1 b_j^+ + \pi_4 + \sum_{j=1}^{n} \pi_k b_j^+ r_d \right) \right\}$$

(34)

$$t_i^j \geq \pi_j^d - d_j \quad \forall j \in I, d \in \Omega$$

(35)

$$t_i^j \geq 0 \quad \forall j \in I, d \in \Omega$$

(36)

$$\sum_{j \in \text{Agn}(e)} t_i^j \leq e \quad \forall d \in \Omega$$

(37)

$$t_i^j \geq 0 \quad \forall j \in I, d \in \Omega$$

(38)

The total tardiness calculations of the partial route $d$ are shown in constraints (35)–(37). So, this problem can be seen as single-vehicle routing problem. In the single machine scheduling sub-problem and single vehicle routing sub-problem, the goal is to find the processing sequence and delivery sequence of orders that produce the lowest corresponding reduced cost, respectively. But solving these two sub-problems is not easy. In this section, two dynamic programming algorithms are used to solve pricing sub-problems.

Dynamic programming for the single-machine scheduling problem: To solve the sub-problems, the dynamic programming approach of Chen and Powell (1999) is used. Consider $A_i$ as the set of orders that are processed before order $j$ and $F(j, c)$ as order $j$, with processing completion time $c$, to minimize the reduced cost. The recursive relation is as follows:

$$F(j, c) = -\pi_j \quad \forall (j = 0, c = 0)$$

(39)

$$F(j, c) = \min_{c \in \Omega} \{ F(i, c) - \pi_i - \pi_j c \} \quad \forall c = 0, \ldots, c_{\text{max}}, j \in I$$

(40)

$$F* = \min \{ F(i, c) \mid c = 0, \ldots, c_{\text{max}}, i \in I \}$$

(41)

First Eq. (39) represents initial state. As it’s clear from this equation, given the constant value of $\pi_j$, there is not any sequence of orders in initial state time 0. In order to do not arrange any order before time 0, an infinite value can be assigned to the reduced cost. In Eq. (40), the reduced cost of partial schedule before order $j$ is denoted by $F(i, c)$ and another term is related to additional created reduced cost after processing of $i$. Also, $c_{\text{max}}$ is maximum completion time. The last Eq. (41) represents the optimal solution of a scheduling sub-problem with the minimum reduced cost. We need to look for an order that satisfies the above equation.

Dynamic programming for single-vehicle routing problem: Let $F^d(i, q)$ as minimum reduced cost of vehicle routing sub-problem where $d$ is a selected route, $t$ is the start delivery time of a vehicle, i is the last order in the batch and $q$ as capacity of vehicle or batch. Also, $B_j$ is set of orders delivered before order $j$. The recursive relation is as follows:

$$F^d(i, q) = F^d + FC - \pi^d \quad \forall (i = 0, q = 0)$$

(42)

$$F^d(i, q) = \min_{\text{all}, (i, j)} \left\{ F^d(i, q - 1) - \pi^d - \pi^j t^j + q \right\} \quad \forall i \in I, \forall t = F_{\text{min}}, \ldots, t_{\text{max}}$$

(43)

$$F* = \min \{ F^d(n + 1, q) \} \quad \forall d, t, q$$

(44)

In the above equations, in the initial state (42), there is not any order to deliver due to the existence constant value, where $d_{\text{max}}$ is maximum delivery time and $F_{\text{min}}$ is smallest processing time of order machine. In order to do not deliver an order before the processing is completed, an infinite value can be assigned to the reduced cost. In Eq. (43), term $F^d(i, q - 1)$ is used as reduced cost of a vehicle before order $i$ and the next terms describes additional reduced cost after adding order in the initial state. Eq. (44) represents the optimal solution of the vehicle routing with minimum reduced cost.

4.4. Branching schemes

After completing the column generation algorithm, if the obtained solution to the RMP problem is not an integer, then we need to get an integer solution using a branching method. After branching, the column generation is executed again on each node of the tree to solve LP relaxation of the problem. This process continues until an integer solution is found or all nodes of the tree are explored. In this case, the branch and price algorithm terminates. Two branching rules are introduced in this part as follows:

Branching on the scheduling variables: To get optimal integer solution of the sub-problem, a branching strategy is designed on $x_{ij}$ variable of the original problem. Let $o_{ij}^e$ equals 1 if order $j$ is processed immediately after order $i$ in schedule $e$ and 0 otherwise. Therefore, $x_{ij} = \sum_{e \in \Omega} o_{ij}^e x_{ij}$. After branching on each node of the branch and bound tree, two child nodes are created. Now, among branches with fractional value, the most fractional value is selected. Then, in one of the branches, the value of 0 is fixed for $x_{ij}$ and in the other branch, the value of 1 is fixed for $x_{ij}$. Each branch will be pruned if the obtained solution is integral or when no improvement will occur in this branch, otherwise each new node is solved by column generation method.

Branching on the routing variables: To solve the problem of the scheduling sub-problem, a branching strategy is designed on $x_{ij}$ variable of the original formulation. Let $h_{ij}^k$ equals 1 if order $j$ is delivered immediately after order $i$ in route $d$ and 0 otherwise. Therefore, $z_{ij}^k = \sum_{e \in \Omega} h_{ij}^k x_{ij}$ for all $k = 1, ..., n$. Two child are created when routing variables are fractional. In the first child $z_{ij}^k$ is fixed by 1 and in the other one $z_{ij}^k$ is fixed by 0 (It means order (i, j) is not assigned to vehicle k). In each created sub-problem initial RMP must be modified, that is, new RMP consists of all columns of parent node except arc (i, j) when $x_{ij}$ or $z_{ij}^k$ is fixed to zero (for vehicle k in the routing problem). If $x_{ij}$ or $z_{ij}^k$ is fixed to one, arcs (i, l) in $A^l \neq j$ and (l, j) in $A_{ij}^l \neq i$ are removed form sub-problem (for vehicle k in the routing problem).
4.5. Lower bound

Calculation of the upper and lower bounds in the branch and bound algorithm is necessary to decide on the continuation of the branching or pruning of each node. A node can be pruned when the obtained lower bound at this node exceeds from the current best upper bound. The upper bound calculation is described in the next section.

By solving linear programming relaxation of the master problem, a lower bound is obtained. Furthermore, by adding subset-row inequalities (SRI), this bound can be more strengthened. Our integrated problem involves two problems: the parallel machines scheduling and the vehicle routing problem (VRP). The first problem can be simulated as a VRP problem, in which machines (or manufacturing sites) are similar to vehicles with predefined numbers. First, orders must be assigned to one of these sites. Then the sequence of processing orders is the same as the delivery sequence in VRP. Jepsen, Petersen, Spoorendonk, and Pisinger (2008) introduce subset-row inequalities for vehicle routing problem with time windows. Let, \( T \) as subset of customer set \( I \), (i.e., \( T \subseteq I \)) and an integer \( \alpha (1 < \alpha < \frac{|I|}{2}) \).

\[
\sum_{d \in n} \left[ \frac{1}{2} \sum_{j \in I} b_{ij} |d_i| \right] \leq \frac{|I|}{\alpha} \quad (45)
\]

These inequalities are also valid for our problem. Suppose that \( \alpha = 4 \) and \( |I| = 5 \). So these inequalities guarantee that at most one route is selected that cover more than one customer in \( T \). The value of lower bound is continuously updated during the algorithm.

4.6. Generating initial columns using Bees algorithm

Logically, the variables which have the ability to improve the objective function or exist in the optimal solution should be at the base. In this study, the generation of initial columns is performed using a metaheuristics algorithm. On the other hand, in the branch and bound method, the use of a proper upper bound can prune many search space that is not promising and can help to improve the computational time of the algorithm. Hence, to select promising columns or generate pool columns, Bees algorithm is used here. The use of heuristics algorithms in exact methods has recently been into consideration. For instance, Kramer, Subramanian, Vidal, and Lucido dos Anjos (2015) use a local search for the Pollution-Routing Problem, or an iterated local search algorithm is used in Cacchiani, Hemmelmayr, and Tricoire (2014) for the periodic vehicle routing problem. In fact, the task of the metaheuristics algorithm is to create a columns pool in which a predefined number of local optimal solutions are stored. Then, using these solutions, the master problem is solved by means of a linear optimization solver.

Bees algorithm is proposed by Pham et al. (2005) inspired by bee’s behavior in finding food sources. First, several bees called scout bees are randomly searched different sites for food sources, and after returning to the hive, they inform the quality of the food source and its distance to the hive by a waggle dance to the other bees. Then, more bees, which are called the recruited bees, are sent to a site with good source of food. The classic version of this algorithm had many parameters that Koc (2010) propose a new version with fewer parameters and improvements in neighborhood search. This algorithm is compatible with combinatorial optimization problems. In this study, a modified version of this algorithm is proposed. For each solution of food source \( j \), a fitness function is defined as \( F_j = \frac{Z_j}{E} \), where \( Z_j \) is objective function of problem. Also, we define \( d_j \) as normalized fitness function.

\[
d_j = \frac{F_j}{F} \quad (46)
\]

Based on the value of \( d_j \), a decision is made to accept or reject the solution \( j \). The probability of quitting a solution for more exploration is calculated as follows:

\[
\pi_j = \begin{cases} 
0.6 & d_j < 0.85 \\
0.2 & 0.85 \leq d_j \leq 1 \\
0.05 & 1 \leq d_j \\
0 & d_j \geq 1.2 
\end{cases} \quad (47)
\]

Therefore, a solution with the higher value of \( d_j \) is more likely to be selected. Fixed values in above equation are derived from the study of Nakrani and Tovey (2004) with a few corrections. If a solution is accepted, two neighborhood operators swap and insertion are used to generate new solutions. In the swap operator, two orders of a solution are selected and their positions are substituted. In the insertion operator, two orders of a solution are selected, and the first order comes to the position immediately after the second order. The stopping criteria for this algorithm is a parameter \( MaxIt \) which is the maximum number of iteration of the algorithm. Moreover, this algorithm has two parameters number of scout bees \( N_s \) and the number of recruited bees \( N_r \). The number of recruited bees of a solution \( j \) has a direct relationship with its fitness function. So, we define another parameter \( \alpha \) instead of \( N_r \), which is defined as \( N_r = \alpha \cdot N_s \cdot d_j \). The pseudo-code of this algorithm is shown in Algorithm 1.

Algorithm 1. Proposed Bees algorithm

5. Computational experiments

The proposed branch and price algorithm and Bees algorithm have been coded in C++ and the relaxed restricted master problem is solved by CPLEX 12.5 software. A personal computer is used for all tests with
2.53 GHz Intel(R) Dual-Core CPU and 4 GB RAM.

5.1. Design of test instances

To perform computational experiments, it is necessary to create a number of problem instances. This is done by defining several combinations of parameters or inputs of the problem. For this study, the number of orders varies from $n = 6$ to 60. There are three cases of sites when $S = 4, 6$ and 8. Moreover, for the processing times, size of orders, due dates and transportation times various uniform distribution are considered as follows:

- $P \sim U(50, 120)$
- $q_i \sim U(4, 8)$,
- $d_i \sim U(\frac{P}{2}, 2 \cdot (P + i))$ and
- $t_i$, $t_i \sim U(250, 600)$.

The fixed transportation cost for a vehicle is 80 and the vehicle capacity limit is 20. Also, the upper bound of total tardiness is set to 600 time unit. Totally, 60 problem instances are generated.

For the branch and price framework, a stopping criterion is required to stop Bees algorithm in each iteration. In this study, a maximum number of iteration is used to stop Bees algorithm. In Bees algorithm, the values of three parameters $Max_{It}$, $N_s$ and $\alpha$ must be tuned. In this study, we obtain these values using the Taguchi test. But, the point is that the value of $N_s$ is the number of generated columns for adding to RMP in each iteration. This column pool size has also been used in Chen and Powell (1999). They have figured out that this size should be between 5 and 10 columns. The lower values lead to the weakening of the results and the higher values lead to an increase in the computational time of the algorithm. Based on this suggestion, a Taguchi experimental design is used to tune the parameters. Three parameters as three factors with three levels lead to a L9 test. The selected levels for parameters are shown in Table 1. A main effects plot for SN ratios is shown in Fig. 3. Therefore, the best values for the parameters are $Max_{It} = 200$, $N_s = 6$ and $\alpha = 0.3$. But on the other hand, the size of columns pool should also be tested on the branch and price algorithm. We perform a test with three levels of 5, 8 and 12 columns for columns pool size in different size 8, 20 and 30 with three sites. The results of this test based on computation times is reported in Table 2. As shown in this table, for the number of orders 8 and 20, the number of six columns had less computational time. On the other hand, for the number order less than 30, the number of six columns and for the order number of more than 30, 10 columns are used.

5.2. Results and analysis

In this part, the obtained results from the proposed branch and price algorithm and its comparison with standalone branch and price algorithm and MILP solver of CPLEX software are reported. Note that, a 3-h time limit is considered for implementation of algorithms. In Tables 3–5, the results of comparing the proposed algorithm with other algorithms are reported based on 4, 6 and 8 number of sites, respectively. The columns information on these tables from left to right are: number of orders ($n$), the number of generated columns to solve master problem ($Columns\#$), the number of nodes exploited in the branch and bound tree ($Nodes\#$) and running time to reach optimal solutions ($R\cdot T.$). In Tables 3–5, the results of three algorithms are reported. In these tables, sign “()” means to reach the maximum limit times, but the algorithm continues to achieve optimal solutions.

As can be seen in the Tables 3–5, in the standalone branch and price algorithm, the average running time values for three tables are 14, 12, and 25 percent greater than the proposed branch and price algorithm and Solutions of CPLEX method are 40, 30 and 42 percent greater than the proposed branch and price algorithm. As shown in Table 3, CPLEX method, even for $S = 4$ sites, cannot solve the problem with 46 orders onward. Standalone B&P also did not solve the last three problems at a given time. But the proposed method could solve all the problems at this time. It is clear from other tables that the proposed method is able to solve the most problems at a given time, and CPLEX method is able to

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Levels</th>
<th>Levels</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Max_{It}$</td>
<td>50</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>$N_s$</td>
<td>3</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Table 1
Levels of parameters of Bees algorithm.

<table>
<thead>
<tr>
<th>Number of orders</th>
<th>Size of columns pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>89.35 86.25 103.96</td>
</tr>
<tr>
<td>20</td>
<td>1563.39 1482.95 1602.65</td>
</tr>
<tr>
<td>30</td>
<td>5095.56 4938.05 4742.91</td>
</tr>
</tbody>
</table>

Table 2
Comparing computation times for different columns pool size.

Fig. 3. Main effects plot for SN ratios.
solve the least problem at this time. The reason for this can be in the structure of the proposed algorithm, create a suitable set of initial solutions by Bees algorithm and using efficient SR cuts.

But on the other hand, the problems discussed above were small size problems. In fact, in the real-world, we face with the larger and more complicated problems. Also, due to the nature of the problem of this paper, which is an operational problem, it needs to be solved daily and sometimes twice a day. So in this section, we will discuss another analysis, in which the solving time limit is set to 40min. If the algorithm does not get the optimal solution at this time, then the algorithm will stop by the user. The resulting gap values form all problem instances are reported in Table 6. This gap is calculated as follows:

$$\text{GAP} (%) = \frac{|Z_{\text{RMP}} - Z_{\text{LRMP}}|}{|Z_{\text{RMP}}|} \times 100$$

where $Z_{\text{RMP}}$ and $Z_{\text{LRMP}}$ are optimal solution of linear relaxation of RMP and integer solution of RMP, respectively. According to this table, the average gap rate for the proposed method is 34% and 62% lower than standalone B&P and CPLEX methods. So the accuracy of the proposed method is better. The value of zero gap on this table indicates the optimal solution is achieved at given time of 40min.

Here, we use statistical analysis to determine whether the differences among the averages of the obtained values of the three methods are different. First, using an ANOVA test with a confidence level of 95%, we compare means of running times of three methods. The output of this test is shown in Table 7. This analysis has been done on all problem instances, and the P-value is greater than 0.05. It means, there is no statistically significant difference between the mean solution time of the three methods. However, one reason for this is that there is a big difference between the minimum and maximum values. In Table 8, an ANOVA test is performed on the gap values among the three methods. P-value is less than 0.05, and so the difference among gap values of the

### Table 3
Results of branch and price algorithm with numbers of site S = 4.

| n  | Proposed branch & price | | | Standalone branch & price | | | CPLEX | | |
|---|---|---|---|---|---|---|---|---|
| 6 | 25 | 3 | 36.13 | 53 | 4 | 42.21 | | |
| 8 | 81 | 5 | 117.64 | 170 | 10 | 138.92 | | |
| 10 | 136 | 23 | 236.53 | 263 | 44 | 250.05 | | |
| 12 | 235 | 47 | 380.46 | 494 | 89 | 444.53 | | |
| 14 | 343 | 93 | 545.15 | 720 | 177 | 611.23 | | |
| 16 | 632 | 158 | 989.19 | 1327 | 300 | 1055.77 | | |
| 18 | 963 | 225 | 1273.53 | 2022 | 447 | 1472.51 | | |
| 20 | 1236 | 296 | 1933.57 | 2596 | 562 | 2083.75 | | |
| 22 | 1569 | 385 | 2329.29 | 3295 | 732 | 2611.63 | | |
| 24 | 2034 | 503 | 2968.06 | 4271 | 956 | 3436.29 | | |
| 26 | 2873 | 786 | 3911.32 | 6033 | 1493 | 4179.51 | | |
| 28 | 3845 | 932 | 4306.16 | 8075 | 1771 | 5343.55 | | |
| 30 | 5068 | 1126 | 5597.96 | 9930 | 2139 | 6297.76 | | |
| 32 | 7654 | 1653 | 6639.00 | 11831 | 3141 | 7112.07 | | |
| 34 | 9331 | 1958 | 7262.70 | 13733 | 3720 | 8183.72 | | |
| 36 | 11,236 | 2319 | 7856.89 | 15634 | 4405 | 10237.83 | | |
| 38 | 13,140 | 2642 | 8467.67 | 17536 | 5015 | 12145.64 | | |
| 40 | 15,045 | 2975 | 9101.66 | 19437 | 5775 | 14307.27 | | |
| 42 | 16,949 | 3308 | 9694.87 | 21339 | 6460 | 16345.02 | | |
| 44 | 18,854 | 3840 | 10,307.78 | 23240 | 7145 | 18529.07 | | |
| 46 | 20,760 | 4568 | 10,911.59 | 25141 | 7832 | 20712.08 | | 
| Average | . | 4195.85 | | | | 4780.26 | | 5911.64 |

### Table 4
Results of branch and price algorithm with numbers of site S = 6.

| n  | Proposed branch & price | | | Standalone branch & price | | | CPLEX | | |
|---|---|---|---|---|---|---|---|---|
| 6 | 48 | 6 | 44.76 | 85 | 11 | 52.30 | | |
| 8 | 154 | 19 | 146.15 | 275 | 19 | 178.27 | | |
| 10 | 239 | 43 | 271.74 | 429 | 88 | 320.88 | | |
| 12 | 447 | 87 | 467.67 | 799 | 179 | 540.45 | | |
| 14 | 652 | 172 | 643.04 | 1167 | 354 | 784.37 | | |
| 16 | 1201 | 292 | 1220.56 | 2149 | 602 | 1354.82 | | |
| 18 | 1830 | 435 | 1634.27 | 3275 | 896 | 1889.62 | | |
| 20 | 2348 | 548 | 2336.73 | 4204 | 1128 | 2673.99 | | |
| 22 | 2981 | 712 | 2898.51 | 5336 | 1467 | 3531.40 | | |
| 24 | 3865 | 931 | 3575.60 | 6918 | 1917 | 4314.04 | | |
| 26 | 5459 | 1454 | 4559.76 | 9771 | 2995 | 5561.90 | | |
| 28 | 7306 | 1724 | 6066.44 | 13077 | 3552 | 6667.15 | | |
| 30 | 9629 | 2083 | 6909.98 | 17236 | 4291 | 7594.13 | | |
| 32 | 14,543 | 3058 | 7645.35 | 26031 | 5700 | 8662.73 | | |
| 34 | 18,343 | 4353 | 8603.02 | 32833 | 8967 | 9947.24 | | |
| 36 | 21,518 | 5578 | 9260.20 | 38516 | 11,490 | 10,752.49 | | |
| 38 | 27,294 | 7631 | 10,238.87 | 48855 | 15,720 | 12,143.25 | | |
| 40 | 33,609 | 10,295 | 12,182.15 | 60160 | 21,204 | 13,287.63 | | |
| 42 | 39,565 | 12,550 | 14,492.36 | 70921 | 25,853 | 14,432.02 | | |
| 44 | 45,611 | 14,908 | (14,943.34) | 81643 | 30,710 | (15,576.40) | | |
| Average | . | 5,353.71 | | | | 4780.26 | | 5911.64 |
three methods are statistically significant. In Fig. 4, the differences among the average gap values of the three methods are shown in the interval plot. So, according to the above analysis, although it was not statistically different among the average running time of the three methods, the results indicate that the proposed algorithm solving time is better than other algorithms. Also, in terms of gaps, the proposed algorithm has the ability to generate tightest solutions.

In Table 9, the results of efficiency tests of the SR inequalities on the proposed branch and price algorithm are shown. A test with six sites is chosen for this part. In Table 9, the first column represents the time taken to achieve an optimal solution without SR inequalities, the second column represents the time taken to achieve optimal solution with SR inequalities and the last column represents number of added SR inequalities in the algorithm. As it's clear from the table, using SR inequalities reduces the computational time and almost 11%, in average, the proposed B&P with SR inequalities could find an optimal solution faster than the B&P without inequalities.

6. Discussion and conclusion

In this study, an integrated decision of production scheduling and distribution in a parallel multi-site supply chain is investigated. Manufacturing sites are far from each other and customers from

![Interval Plot of Proposed B&P, Standalone B&P, CPLEX](Fig. 4. Interval plot for three methods.)
different geographical locations place their orders to supply chain management.
Each order is allocated to just one site based on the distance of sites and customers. After completion of orders processing in each site, they have to be distributed among customers. A batch delivery consideration with routing decisions is assumed in this study. A two-agent view is presented in which total tardiness and total cost of transportation are considered as objectives of the first and second agents.

A mathematical formulation is developed for this problem. Due to a large number of integer variables, a column generation based approach is used as the solution approach. First, using Dantzig-Wolfe decomposition, the original formulation is decomposed to master problem and two sub-problems. Then a branch and price framework is introduced to solve this problem. In this approach, a Bees algorithm is used to generate initial solutions in each iteration of column generation algorithm. Sub-problems are solved via dynamic programming and two branching rules are presented to obtain integer optimal solutions. The proposed approach is compared with a MILP solver of CPLEX and standalone branch and price algorithm. Some computational tests are executed in different problem instances which are solved by proposed algorithm. The results show that although the proposed algorithm leads to an increase of computational time than the standalone branch and price algorithm, the algorithm is able to produce tight bounds.

There are different directions for future researches. In the problem aspect, other assumptions can be used such as different scheduling criteria, different routing strategy or different environments (serial or network). Based on solution approach, other exact methods can be used for this problem. Different methods to solve pricing problem such as labeling methods or different branching approach can be used for the branch and price method. Due to the complexity of the problem, other heuristics or metaheuristics methods can be applied to the problem.

Declaration of Competing Interest
We have no conflicts of interest to disclose and sources of support.

References
Agnetis, A., Merchanpandi, P. B., Pacciarelli, D., & Pacifici, A. (2004). Scheduling problems for this problem. Different methods to solve pricing problem such as heuristics or metaheuristics methods can be applied to the problem.

Table 9
The effect of SR inequalities on the proposed B&P approach.

<table>
<thead>
<tr>
<th>n</th>
<th>R. F.</th>
<th>R. p,SR</th>
<th>#SR</th>
<th>n</th>
<th>R. F.</th>
<th>R. p,SR</th>
<th>#SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>52.37</td>
<td>44.76</td>
<td>4</td>
<td>26</td>
<td>5334.91</td>
<td>4559.7552</td>
<td>66</td>
</tr>
<tr>
<td>18</td>
<td>160.07</td>
<td>146.15</td>
<td>4</td>
<td>38</td>
<td>6967.47</td>
<td>6006.44</td>
<td>91</td>
</tr>
<tr>
<td>10</td>
<td>282.60</td>
<td>271.74</td>
<td>4</td>
<td>30</td>
<td>8084.67</td>
<td>6909.9756</td>
<td>43</td>
</tr>
<tr>
<td>12</td>
<td>491.05</td>
<td>467.67</td>
<td>32</td>
<td>34</td>
<td>9718.87</td>
<td>7649.0568</td>
<td>98</td>
</tr>
<tr>
<td>14</td>
<td>758.79</td>
<td>643.04</td>
<td>19</td>
<td>38</td>
<td>10151.56</td>
<td>8603.0208</td>
<td>96</td>
</tr>
<tr>
<td>16</td>
<td>1367.03</td>
<td>1220.56</td>
<td>36</td>
<td>42</td>
<td>10556.62</td>
<td>9260.196</td>
<td>82</td>
</tr>
<tr>
<td>18</td>
<td>1699.64</td>
<td>1634.27</td>
<td>54</td>
<td>46</td>
<td>10546.04</td>
<td>10238.872</td>
<td>77</td>
</tr>
<tr>
<td>20</td>
<td>2733.97</td>
<td>2336.73</td>
<td>47</td>
<td>50</td>
<td>12913.08</td>
<td>12182.1524</td>
<td>99</td>
</tr>
<tr>
<td>22</td>
<td>3478.21</td>
<td>2898.51</td>
<td>32</td>
<td>54</td>
<td>14339.99</td>
<td>13482.3632</td>
<td>132</td>
</tr>
<tr>
<td>24</td>
<td>4254.96</td>
<td>3575.5984</td>
<td>85</td>
<td>60</td>
<td>15652.32</td>
<td>14943.3414</td>
<td>139</td>
</tr>
</tbody>
</table>

Average | 5950.61 | 5353.71 |
