Journal of Quality Engineering and Production Optimization

Vol. 5, No. 1, Winter & Spring 2020

http://jqepo.shahed.ac.ir

Research Paper

DOI: 10.22070/JQEPO.2020.3173.1060

A Combinatorial Benders Cut for the Integrated Production Scheduling and Distribution Problem

Ali Gharaei [1] , Fariborz Jolai [1*]

[1] *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

**** Corresponding Author:** Fariborz Jolai (Email: fjolai@ut.ac.ir)*

Abstract *– One of the most attractive topics for industry and researchers in industrial engineering is the integration of decisions in the supply chain. There are some advantages in the integrated decisions compared to different decisions, such as decreasing the cost of distribution and On-Time delivery. An integrated production scheduling and distribution problem is discussed in this study. The main contribution of this paper is to study this problem from a multi-agent viewpoint. In this case, each agent has a set of customers with their jobs, and each agent has a specific objective. Here, a two-agent problem is discussed in which the first agent objective is the minimization of the total tardiness for jobs of the first agent and the second agent objective is to minimize the total cost of the distribution. For this problem, a mixed-integer linear programming formulation is developed. Due to the complexity of the original problem and its specific structure, a combinatorial Benders decomposition approach decomposes it to the master problem and sub-problem. It means some modifications have been applied to the classic version of Benders method. The results show the excellent performance of the algorithm in comparison with another exact method .*

Keywords*– Batch delivery, Multi-Agent scheduling, Routing decision, Benders decomposition.*

I. INTRODUCTION

There are numerous decisions in the supply chain to be taken, such as inventory management, logistics, production scheduling, distribution decisions, etc. Two latter decisions are discussed in this paper. Both of these decisions are made at the operational planning level. Heretofore, in the supply chain, each member of the supply chain tries to plan and schedule their jobs and maximize its profit, regardless of other members. Nevertheless, nowadays, the coordination and integration of decisions among each supply chain member are critical. In recent years several papers are addressed this problem, such as Chandra and Fisher (1994), Chen and Vairaktarakis (2005), Lei et al. (2006), and Bard and Nananukul (2009). In these papers, they are shown that the integrated scheduling and distribution decisions can considerably reduce operations costs and improve customer satisfaction by On-Time delivery.

Various distribution strategies are introduced in the literature, such as vehicle routing and cross-docking. A key point in these strategies is the aggregation of products called the batch delivery method. In this approach, multiple jobs from one or different customers, after processing, are allocated to a batch or a vehicle, and then they are transferred to the customers. Batch delivery, in comparison to individual delivery, considerably reduces the costs of transportation.

Manuscript Received:30-Oct-2019 & Revised: 25-Jan-2020 & Accepted: 16-March-2020
ISSN: 2423-3781

According to Cheng and Kahlbacher (1993), the batch delivery problem concerns scheduling some jobs on certain machines or factories that are supposed to consolidate and dispatch to several downstream factories or customers in batch by vehicles. Various objectives are considered for this problem. The main objective is the scheduling criteria such as makespan, total tardiness, and so on. However, distribution cost is another essential objective that is used in papers with distribution consideration. Moreover, total tardiness of jobs and total distribution cost, which are considered in this study, are conflicting if both of them are considered in a problem simultaneously. Since to reduce the distribution cost, the number of batches has to be minimized that, in turn, requires filling the capacity of each vehicle.

On the other hand, this can increase the total tardiness and costs of warehousing and customer dissatisfaction. Hence, a balance between distribution costs and scheduling criteria is required. It is also possible to send jobs to multiple customers by a vehicle, so routing decisions can be applied among customers.

Researchers in the scheduling field try to get closer to their problems with real-world problems. One of the most recent of these assumptions is considering multiple agents for a scheduling problem. In classical multi-criteria scheduling problems, it is assumed that all jobs are scheduled based on one or more objective(s). For example, Li et al. (2016) discussed a multi-objective integrated production and distribution scheduling problem: distribution cost and the total customer waiting time. However, in real problems, maybe multiple sets of jobs exist with their objective(s) called the multi-agent scheduling problem. In Multi-Agent Scheduling, instead of a single set of jobs with a common goal or goals, several groups of jobs have a distinct objective. One agent owns each of these job groups. Now, given the available facilities in the workplace, the jobs must be processed on shared machines to satisfy each working group (agents)' objectives. In fact, in multi-agent scheduling problems, each agent uses common resources according to its objective and compete with each other to achieve these resources. In recent years, many studies have been studied in this field. The pioneer of this field is Baker and Cole Smith (2003) and Agnetis et al. (2004), which in their papers, the multi-agent scheduling problem had been introduced.

From a structural point of view, multi-agent problems are divided into four categories: Competing agent, Interfering set, Multi-criteria, and Non-disjoint Sets. In the first case, the agents have no typical job, and each job belongs to only one agent. In the second case, job sets are nested so that a set may be the subset of other sets. For example, in a two-agent case, a set is a subset of the entire jobs. The third is the classic multi-criteria scheduling problem, in which all jobs have the same objectives. Finally, the fourth case is the general case, in which each job can belong to one or more sets. Also, there are three common approaches to solve multi-agent scheduling problems. The Constrained optimization problem is a widely used method in which one objective is minimized, and the other ones are bounded. In the linear combination method, all objectives are combined linearly and become a single-objective problem. Finally, in the Pareto optimization approach, the goal is to find the whole set of strict Pareto optimal solutions (Agnetis et al. (2014)).

Here, a scheduling and distribution integration problem is studied, and the routing decisions among customers are considered. To give an example, in the real world, consider a supply chain where customers from different locations send orders to Supply chain management. Supply chain management tries to schedule jobs and to plan for distribution jobs over a short period. The production policy is based on MTO. Another innovation in this study is studying the problem from the multi-agent view. The first agent is some customers whose objective is to minimize the total tardiness, and the second agent's objective is to minimize total distribution costs for all jobs. In this study, the multi-agent problem is referred to interfering jobs case, and a constrained optimization approach is used to deal with this problem. That is, total distribution cost is considered as objective and total tardiness of the first agent jobs is bounded. A mixed-integer programming model is proposed for the problem. Due to the complicated structure of the problem, a modified Benders decomposition is used to solve the model. Due to the complexity of the original problem structure, the original problem turns to the master problem and sub-problem, and then in the iterative procedure, solutions are found.

The rest of the paper is as follows: Section II reviews papers related to the proposed problem and how to solve it. In Section III, the problem is described in detail, and then the problem is formulated. In Section IV, a modified Benders algorithm is presented to solve the problem. Computational results of the paper used to evaluate the efficiency of the proposed algorithm are reported in Section V. Conclusions and some suggestions for future research are stated in the last section.

II. RELATED WORK

In recent years, there are several published papers in the fields of integrated scheduling and distribution, batch delivery problem, and the multi-agent scheduling, which are reviewed in this section. One of the most attractive topic in these years is the integration of production scheduling and distribution decisions, that is our main concern in this study. Chandra and Fisher (1997) investigate the coordination of production and distribution planning problems and compare two cases of the separated and integrated decision. In this paper, a factory with multiple products is addressed, which in the distribution part, vehicles with unlimited capacity and split delivery are considered. They show that the operation cost could be reduced from 3% to 20% in the integrated case. Also, some other papers such as Chen and Vairaktarakis (2005) and Pundoor and Chen (2005) are published to investigate the profit of using the optimal integrated production-distribution schedule compared to the separated decisions.

In recent years, He et al. (2019) studied integrated production and distribution operations with multiple plants, multiple order sizes, and multiple transportation modes in the commit-to-delivery business mode, aiming to minimize the total cost production and distribution. They develop a memetic algorithm for this problem. Gosh and Mondal (2018) discussed a case study of integrated production-distribution planning in the dairy industry. A mixed-integer linear programming (MILP) model is developed to maximize the overall profit contribution of the business. The MILP is solved by using CPLEX software. Kesem and Bektas (2019) discuss an integrated production and outbound distribution scheduling problem along with time windows in a supply chain. The machine setting in the supply chain is based on a parallel machine. Marandi and Fatemi Ghomi (2019) proposed an integrated production and distribution scheduling problem in a multi-factory supply chain. They developed an improved Imperialist Competitive Algorithm (ICA) along with a local search based on the simulated annealing algorithm. In Ji et al. (2019), an MILP formulation is developed for an integrated model for the production-inventory-distribution problem in an interconnected open global logistics network.

Different authors consider different objective functions for the integrated scheduling and distribution problem. For example, in some papers, the tradeoff between the cost of transportation and the customer satisfaction performance measure are considered such as Hall and Potts (2003), Hall and Potts (2005), Chen and Vairaktarakis (2005), Wang and Lee (2005), Li and Vairaktarakis (2007) and Pundoor and Chen (2005). On the other hand, in some papers, the tradeoff between the cost of transportation and total inventory cost is addressed, such as Herrmann and Lee (1993), Lee and Chen (2001), Geismar et al. (2008), and Li et al. (2005). Also, some papers study customer service levels regarding vehicle availability constraints (e.g., Stecke and Zhao (2007) and Pundoor and Chen (2009)). Some paper investigates the effect of integration for time-sensitive products. In Garcia and Lozano (2005) and Garcia and Lozano (2004), a perishable product such as ready-mix concrete is described, and industrial adhesive materials are used in Geismar et al. (2008) and Armstrong et al. (2008). Here, we consider total distribution cost & total tardiness as objective functions.

One of the issues we cover in this study is the batch delivery in the distribution part. For the first time, Cheng and Kahlbacher (1993) defined each batch delivery date to the distribution system as equals the time it takes to complete the last job in the batch. They used a branch and bound approach and provided lower and upper bounds for this problem. Cheng et al. (1996) discussed single machine scheduling with batch delivery so that the total number of batches and earliness penalty are minimized. Hall and Potts (2003), as one of the most cited articles on batch delivery, address scheduling jobs on a single machine, with batch delivery consideration and different objectives, such as total flow time, maximum tardiness, and the number of tardy jobs. They developed a dynamic programming algorithm to minimize these objectives, as well as distribution costs. A problem with some jobs on a parallel machine and then the batch delivery approach is discussed in Wang and Cheng (2003), such that the total flow time and delivery costs are reduced. They have shown that the complexity class of the problem is NP-complete even for two machines, and a dynamic programming algorithm is used to solve the problem. Selvarajah et al. (2013) focus on the single machine scheduling problem with batch delivery and release times such that the sum of weighted flow times and delivery costs are minimized, and then an approximation algorithm is presented for in equal weight case. They also provide a metaheuristic method for the general problem. In Yin et al. (2013), a single-machine scheduling problem is investigated, and batch delivery with an assignable common due date and controllable processing times is addressed. Recently, a rule-based meta-heuristics is developed for unrelated parallel machine scheduling and batch delivery by Joo and Kim (2017) to minimize the makespan of the whole process.

In this study, a multi-agent viewpoint on the integrated scheduling and distribution problem is introduced. The multi-agent scheduling problems are reviewed and categorized by Perez-Gonzalez and Framinan (2014). A framework for future research is shown for this problem and solution approaches. They discuss applications of the multi-agent scheduling, which is related to the supply chain scheduling, such that the total cost of production and distribution is minimized, and customers compete to achieve supply chain resources. Agnetis et al. (2004) present some examples of different multi-agent scheduling cases with a constrained optimization approach. In the two-agent case, a branch and bound algorithm and Lagrangian relaxation for the single machine scheduling problem are developed by Agnetis et al. (2009). The objectives are minimizing the weighted total completion time of the first agent jobs, and the second agent objective, as the form of upper bound, is the cost function of the second agents' jobs. Rohmer and Billaut (2015) investigate the problem of integrated scheduling and distribution in the two-agent case. The first agent is a factory with a flow-shop environment, and another is the 3PL distributor. The factory problem involves minimizing inventory costs, equipment, and tardiness penalties, and in the 3PL problem, the objective is minimizing the costs of route and delays that have to be paid to the manufacturer. Qi Tan et al. (2011) developed an ACO algorithm for a two-agent single machine scheduling problem with a makespan objective function for both agents. Yin et al. (2016) present a two-agent single-machine scheduling model for the batch delivery problem, and a constrained optimization approach is used to solve the problem. Complexity is discussed, and a polynomial-time algorithm is developed to solve the problem. Lin et al. (2017) developed a particle swarm optimization to solve the multi-facility customer order scheduling problem, and Yin et al. (2015) used a honey-bees optimization algorithm to solve a two-agent single-machine scheduling problem with ready times.

Some research gaps have been found by reviewing the literature. Most of the papers in the batch delivery literature focus on direct shipping of products to customers. Little work has been done on the case of multi-customer and distribution strategies such as vehicle routing or cross-docking. Additionally, no paper has a multi-agent perspective on this problem. Contributions of this study are summarized as follows:

- Develop a mathematical formulation for the batch delivery problem
- Consider the problem in the case of multiple customers along with routing decisions between customers
- Present a multi-agent view of the problem
- Develop a Benders decomposition approach to achieve optimal solutions

III. PROBLEM DESCRIPTION

Suppose that a supply chain that receives customers' orders at the beginning of the planning horizon and after processing jobs, must be distributed among customers who are in different geographic locations. Each job has a predetermined due date, and jobs have to be consolidated for delivery. It means several jobs form a batch for distribution by a vehicle. There are routing decisions among customers. At first, the sequence of jobs in the supply chain should be specified. Then, each job has to be assigned to a vehicle. Finally, the order of jobs for delivery to customers has to be determined. This is a multi-agent problem with interfering jobs set. So, the total distribution cost has to be minimized, and the total tardiness of the first agent’s jobs is bounded.

A. Assumptions

A set of assumptions fully describes each problem. Assumptions of this study are as follows:

- In the scheduling part, the preemption is not allowed
- All jobs are available at the beginning of the planning horizon
- There is no set-up time before processing, and loading and unloading times are included in shipping time
- Processing machines and vehicles are always available at the factory
- The batches, or vehicles, are have limited capacity
- The fleet is homogeneous and sufficient vehicles are available

B. Notation

In this section, we introduce the notation used in this paper:

Sets

J Set of jobs or customers $i, j \epsilon \{1,2, \dots, n\}$

K Set of vehicles $k \epsilon \{1,2, \dots, \text{n}\}$

Parameters:

n Total number of jobs or customers

M A positive big number

p_i Processing time of job *i*

d_i Due date of job *i*

q_i Size of job *i*

t_{ij} Transportation time between customers *i*th and *j*th

t_{0i} Transportation time between customer *i*th and the factory

r_{ij} Cost of transportation between customers *i*th and *j*th

r_{0i} Cost of transportation between customer *i*th and the factory

FC Fixed cost of each vehicle transportation

Q Vehicle capacity

δ The upper bound of total tardiness, determined by the first agent

Decision variables

X_{ij} Equals 1 if job j processed immediately after job i and equals 0 otherwise

Y_j^k Equals 1 if job j is assigned to the *k*th batch (vehicle) and equals 0 otherwise

Z_{ij}^k Equals 1 if vehicle k transfer job *j* immediately after job *i and* equals 0 otherwise

C_j Completion time of the job *j*

Cb_k Completion time of the batch k (equal to departure time of vehicle k)

T_j Tardiness of the job

A_j Delivery time of job j

C. Mathematical formulation

Two dummy jobs 0 and n+1 are introduced with zero processing times and due dates for determining the first and the last job of the sequence. Also, two customers and n+1 are introduced with zero transportation cost, which zero customer is the factory, and n+1 customer means returning to the manufacturing site. Now, the problem formulation described as bellows:

- This set of constraints, determine the sequence of jobs in the factory and the first and last jobs of the sequence. It means a job is either the first job or the last job in the factory for processing. Otherwise, there is a job before, and a job after it is for processing.

$$\sum_{\substack{i=0 \\ i\neq j}}^{n} X_{ij} = 1 \qquad \forall j = 1,2,\dots,n+1 \tag{1}$$

$$\sum_{\substack{j=1 \\ i\neq j}}^{n+1} X_{ij} = 1 \qquad \forall i = 0,1,\dots,n \tag{2}$$

$$X_{0,n+1} = 0 \tag{3}$$

$$X_{ij} + X_{ji} \leq 1 \qquad \forall i,j = 1,2,\dots,n \tag{4}$$

- Assignment of jobs to vehicles or batches is shown as follows:

$$\sum_{k=1}^{n} Y_j^k = 1 \qquad \forall j = 1,2,\dots,n \tag{5}$$

- The following constraints refer to the sequence of customer deliveries. It means a job is either the first job or the last job in the delivery vehicle. Otherwise, there is a job before, and a job after it is for delivery:

$$\sum_{\substack{i=0 \\ i\neq j}}^{n} Z_{ij}^{k} = Y_j^k \qquad \forall j = 1,2,\dots,n+1 \quad , \quad \forall k = 1,2,\dots,n \tag{6}$$

$$\sum_{\substack{j=1 \\ i\neq j}}^{n+1} Z_{ij}^{\,k} = Y_i^k \qquad \forall i = 0,1,\ldots,n \quad , \quad \forall k = 1,2,\ldots,n \tag{7}$$

$$Z_{0,n+1}^{\,k} = 0 \qquad \forall k = 1,2,\ldots,n \tag{8}$$

$$Z_{ij}^{\,k} + Z_{ji}^{\,k} \leq 1 \qquad \forall i,j,k = 1,2,\ldots,n \tag{9}$$

- This equation sets the balance of arrival and departure of each customer node in the vehicle routing problem; (i.e., for a customer, such as h, once the vehicle is entered and once is exited)

$$\sum_{\substack{i=0 \\ i\neq h}}^{n} Z_{ih}^k - \sum_{\substack{j=1 \\ h\neq j}}^{n+1} Z_{hj}^k = 0 \qquad \forall h,k = 1,2,\ldots,n \tag{10}$$

- Capacity limits for vehicles are shown as bellows:

(It means the sum of jobs allocated to a vehicle should not be greater than the capacity of the Q. The final term states that if it is not assigned any job to the vehicle, the vehicle capacity will be zero.)

$$\sum_{j=1}^{n} q_j Y_j^k \leq Q * \sum_{j=1}^{n} Z_{0j}^k \qquad \forall k = 1,2,\ldots,n \tag{11}$$

- Calculation of the completion times of jobs and vehicles are:

$$C_j \geq P_j - M\left(1 - X_{0j}\right) \qquad \forall j = 1,2,\ldots,n \tag{12}$$

$$C_j \geq C_i + P_j - M\left(1 - X_{ij}\right) \qquad \forall\, i,j = 1,2,\ldots,n \tag{13}$$

$$Cb_k \geq C_j - M\left(1 - Y_j^k\right) \qquad \forall j = 1,2,\ldots,n \quad ,\forall k \tag{14}$$

- Calculation of delivery times (Also, is a sub-tour elimination constraint)

$$A_j \geq Cb_k + t_{0j} - M\left(1 - Z_{0j}^k\right) \qquad \forall j = 1,2,\ldots,n \quad ,\forall k \tag{15}$$

$$A_j \geq A_j + t_{ij} - M\left(1 - Z_{ij}^k\right) \qquad \forall\, i,j = 1,2,\ldots,n \quad ,\forall k \tag{16}$$

- Jobs tardiness constraints are calculated as follows:

$$T_j \geq A_j - d_j \qquad \forall j = 1,2,\ldots,n \tag{17}$$

$$T_j \geq 0 \qquad \forall j = 1,2,\ldots,n \tag{18}$$

- the objective function for the jobs of the first agent has been bounded:

$$\sum_{j\epsilon\, Agent1}^{n} T_j \leq \delta \tag{19}$$

- integrality and non-negativity constraints

$$X_{ij}, Y_j^k, Z_{ij}^{\ k} \in \{0,1\}$$
$$C_j, Cb_k, A_j, t_{ij} \geq 0 \quad (20)$$

- The objective function is as follows:

$$\min Z = \sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij} Z_{ij}^{\ k} + \sum_{k=1}^{n}\sum_{i=1}^{n} r_{i0} Z_{i,n+1}^{\ k} + \sum_{k=1}^{n}\sum_{j=1}^{n} r_{0j} Z_{0,j}^{\ k} + FC * \sum_{j=1}^{n}\sum_{k=1}^{n} Z_{0,j}^{\ k} \quad (21)$$

The first term refers to transportation costs between customers. The second and third terms refer to transportation costs between the factory and the first customers in each vehicle and between the last customers in each vehicle and the factory. The last term refers to the fixed cost of transportation of each vehicle.

IV. DECOMPOSITION APPROACH

A. Classic Benders decomposition

It is often tough to solve a problem with an exact method because there is more than one decision simultaneously, such as the position of the machine and allocation of jobs, allocation of jobs to vehicles, and routing decisions. Each of these decision-making cases may not be complicated by itself, but considering all of them simultaneously in one problem is not easy. An innovative method is the decomposition approach and can be useful in such a problem. In this method, the problem is decomposed into two or more problems (assumed to be independent), and each one is solved separately; then, a method for coordinating and combining these partial solutions is used to reach the optimal solution. In the decomposition approach, sub-problem (which are generally linear and straightforward problems) are solved in each iteration with different parameters, and the information obtained from them is used in the master problem. The output of the master model is the input parameters of the sub-problems. In each repetition, after solving the sub-problems and obtaining their output, we decide on the optimality of the original problem or continue the iteration of the algorithm.

Benders proposed the Benders decomposition algorithm in 1962 for MIP and IP problems based on the idea of problem decomposition and cutting plane. Benders decomposition is a well-known approach that is used when dealing with hard variables. In the Benders approach, based on dual theory, we divide the primary optimization problem into two problems: the master problem and the sub-problem. First, we find a solution to the master problem, which is the same as the original problem without constraints or very few constraints. Then, by using the master problem solution, we run the sub-problem that remains a benchmark for the fulfillment of the constraints. If the constraints are met, the solution is optimal. Otherwise, we add the constraints that have the most severe violations with the help of the Benders optimality and feasibility Cut, and we will repeat this procedure to satisfy all constraints. Usually, this method does not require solving the sub-problem, but its dual problem (DSP) is used.

The termination condition of the algorithm is based on the optimality gap, which is calculated as $|Z_{up} - Z_{down}| \leq \varepsilon$. After solving the master model, we can calculate the lower bound for the minimization problem. At each step, we have $Z_{down} = Z_{Master}$. Namely, by using the master model solution, the objective function value represents the lower bound. After solving the DSP model, one can also calculate the upper limit in the Minimization problem. At each step, we have $Z_{up} = Z_{DSP} + f\ (\bar{x}^{\nu})$. Namely, having the master problem solution ($\bar{x}^{\nu}$) and the objective function of the DSP problem, the value of Z_{DSP}, plus the expression value, contains the problematic variable ($\bar{x}^{\nu}$) shows the upper limit.

B. Structure of the problem

An appropriate applying decomposition approach needs to discover the problem structure. So, the original problem is decomposed into a master problem and sub-problem. Here, variables y and z are the master problem variables, and variable x is the sub-problem variables, and all of them are binary variables. The objective function of the master problem is the same as the original problem. The sub-problem has not objective function, and so it is a feasibility problem. In this study, the scheduling problem is considered the sub-problem because it is less complicated than the routing problem. But this problem has integer variables. So, in this case, it is not possible to implement the classic Benders method since the duality theory used for linear problems is not applicable to the integer problems. Therefore, we combine the general framework of classical Benders decomposition, but in the cut generation phase, we use a combinatorial Benders cut, which will be described in the next section. Also, some adjustments are conducted regarding the structure of the problem under study. There is also an approach to deal with big-M constraint. The sub-problem is as follows:

$$\sum_{\substack{i=0 \\ i\neq j}}^{n} X_{ij} = 1 \qquad \forall j = 1,2, \dots, n+1$$

$$\sum_{\substack{j=1 \\ i\neq j}}^{n+1} X_{ij} = 1 \qquad \forall i = 0,1, \dots, n$$

$$X_{0,n+1} = 0$$

$$X_{ij} + X_{ji} \leq 1 \qquad \forall i,j = 1,2, \dots, n$$

$$C_j \geq P_j - M(1 - X_{0j}) \qquad \forall j = 1,2, \dots, n$$

$$C_j \geq C_i + P_j - M(1 - X_{ij}) \qquad \forall\, i,j = 1,2, \dots, n$$

$$Cb_k \geq C_j - M(1 - \bar{y}_j^k) \qquad \forall j = 1,2, \dots, n \quad ,\forall k$$

$$A_j \geq Cb_k + t_{0j} - M(1 - \bar{Z}_{0\,j}^k) \qquad \forall j = 1,2, \dots, n \quad ,\forall k$$

$$A_j \geq A_j + t_{ij} - M(1 - \bar{Z}_{ij}^k) \qquad \forall\, i,j = 1,2, \dots, n \quad ,\forall k$$

$$T_j \geq A_j - d_j \qquad \forall j = 1,2, \dots, n$$

$$T_j \geq 0 \qquad \forall j = 1,2, \dots, n$$

$$\sum_{j\epsilon\, Agent1}^{n} T_j \leq \delta$$

$$X_{ij} \in \{0,1\}$$

$$C_j, Cb_k, A_j, t_{ij} \geq 0$$

where $\bar{y}_j^k$ and $\bar{Z}_{ij}^k$ are the fixed values obtained from the optimal solution of the master problem? In the Benders decomposition approach, using Big-M constraint is not desirable. Thus, the value of M must be in the smallest possible of value. So, we choose these values for Big-M constraints:

- For constraints (14) , (15) and (16), $M = \sum_{j=1}^{n} P_j$
- For constraints (17) and (18), $M = \sum_{j=1}^{n} P_j + \sum_{j=1}^{n}\sum_{i=1}^{n} t_{ij} + \max t_j$

The master problem is as follows:

$$\min Z = \sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij} Z_{ij}^{k} + \sum_{k=1}^{n}\sum_{i=1}^{n} r_{i0} Z_{i,n+1}^{k} + \sum_{k=1}^{n}\sum_{j=1}^{n} r_{0j} Z_{0,j}^{k} + FC * \sum_{j=1}^{n}\sum_{k=1}^{n} Z_{0,j}^{k}$$

$$\sum_{k=1}^{n} Y_j^k = 1 \qquad \forall j = 1,2,\dots,n$$

$$\sum_{\substack{i=0,\\ i\neq j}}^{n} Z_{ij}^{k} = Y_j^k \qquad \forall j = 1,2,\dots,n+1 \quad , \quad \forall k = 1,2,\dots,n$$

$$\sum_{\substack{j=1\\ i\neq j}}^{n+1} Z_{ij}^{k} = Y_i^k \qquad \forall i = 0,\dots,n \quad , \quad \forall k = 1,2,\dots,n$$

$$Z_{0,n+1}^{k} = 0 \qquad \forall k = 1,2,\dots,n$$

$$Z_{ij}^{k} + Z_{ji}^{k} \leq 1 \qquad \forall i,j,k = 1,2,\dots,n$$

$$\sum_{i=0}^{n} Z_{ih}^{k} - \sum_{j=1}^{n+1} Z_{hj}^{k} = 0 \qquad \forall i,j,k = 1,2,\dots,n$$

$$\sum_{j=1}^{n} q_j Y_j^k \leq Q * \sum_{j=1}^{n} Z_{0j}^{k} \qquad \forall k = 1,2,\dots,n$$

$$Y_j^k, Z_{ij}^{k} \in \{0,1\}$$

C. Combinatorial Benders cuts(CBCs)

Codato and Fischetti (2006) present a version of Benders decomposition which is used for problems whose objective function is only contains had variables and has a linear sub-problem problem. This approach aims to reduce the complexity of the master problem by reducing the number of integer variables. The complexity of the master problem is reduced, and the complexity of the sub-problem is increased. Therefore, rather than having a master problem defined over the full set of integer variables, the new decomposition approach formulates a master problem defined over a subset of the integer variables. In contrast, the remaining integer variables are transferred to the sub-problem. In what follows, we present a general description of combinatorial Benders decomposition for mixed integer programming (MIP) problems with the following structure:

$$\begin{aligned} [OP]: \quad & Z_{OP} = \min C^T x + D^T y \\ s.t. \quad & Ax + By \geq e_1, \\ & Ex \geq e_2, \\ & Fy \geq e_3, \\ & x_i \in \{0,1\}, \quad y_j \geq 0 \end{aligned}$$

Where x is the complex variable, in contrast to classical Benders decomposition, in which the sub-problems are linear programs, combinatorial Benders decomposition leaves some of the integer variables in the sub-problem, leading to a MIP sub-problem. For fixed $\bar{x}$ variables, the original problem leads to the sub-problem:

$$
\begin{aligned}
[SP]: \quad & Z_{SP}(\bar{x}) = \min D^T y \\
s.t. \quad & Ey \geq e_1 - A\bar{x}, \\
& Fy \geq e_3, \\
& y_j \geq 0
\end{aligned}
$$

and the master problem is:

$$
\begin{aligned}
[MP]: \quad & Z_{MP} = \min C^T x + \theta(x) \\
s.t. \quad & Ex \geq e_2, \\
& x_i \in \{0,1\}
\end{aligned}
$$

Suppose that $D = 0$, which is the objective function of the original problem, include only complicated or master problem variables. In this case, like our discussed problem, there is no objective function for the sub-problem, and this problem turns to the feasibility problem. Therefore, the sub-problem role is only checking the feasibility or infeasibility of the master problem solution. In the feasibility problem, if Y is the solution space and $Y \neq 0$, all points in the $y \in Y$ are feasible and optimum.

At first, the Master problem has to be solved. If the master problem is infeasible, the original problem is also infeasible and otherwise, if the master problem is feasible, let $\bar{x}^\nu$as an optimum solution of the master problem in iteration ν and put it in the sub-problem. Now, if $SP^\nu(\bar{x}^\nu)$ is feasible, then $(\bar{x}^\nu, \bar{y}^\nu)$ is the optimum solution to the original problem. But if $\bar{x}^\nu$ turns the $SP^\nu(\bar{x}^\nu)$ to the infeasible problem, the OP is also infeasible, and the following feasibility cut has to be added to the master problem.

$$
\sum_{i \,\forall\, x_i^\nu = 0} x_i + \sum_{i \,\forall\, x_i^\nu = 1} (1 - x_i) \geq 1
$$

This cut is called *Combinatorial Benders' cut (CBC),* and it causes to change at least one of the components of $\bar{x}^\nu$. One or more CBCs of this type are generated in correspondence with a given infeasible $\bar{x}^\nu$, and added to the master problem. On the other hand, if the $SP^\nu(\bar{x}^\nu)$ is feasible; the following optimality cut has to be added to the master problem.

$$
M\left(\sum_{i \,\forall\, x_i^\nu = 0} x_i + \sum_{i \,\forall\, x_i^\nu = 1} (1 - x_i)\right) + \theta \geq q^\nu
$$

where q^ν is the objective function of the optimal solution in the iteration ν. In each iteration, after adding the combinatorial Benders' cut or optimality cut to the master problem, a new iteration is formed, and again, the steps are repeated by solving the master problem. So, the master problem is as follows:

$$
\begin{aligned}
[MP]: \quad & \min C^T x + \theta \\
s.t. \quad & M\left(\sum_{i \,\forall\, x_i^\nu = 0} x_i + \sum_{i \,\forall\, x_i^\nu = 1} (1 - x_i)\right) + \theta \geq q^\nu \qquad \forall P \\
& \sum_{i \,\forall\, x_i^\nu = 0} x_i + \sum_{i \,\forall\, x_i^\nu = 1} (1 - x_i) \geq 1 \qquad \forall R \\
& \theta \geq 0, \; x_i \in \{0,1\}
\end{aligned}
$$

where P and R are the optimality cut and the combinatorial Benders' cut. In Fig. (1), the structure of the solution approach is depicted.

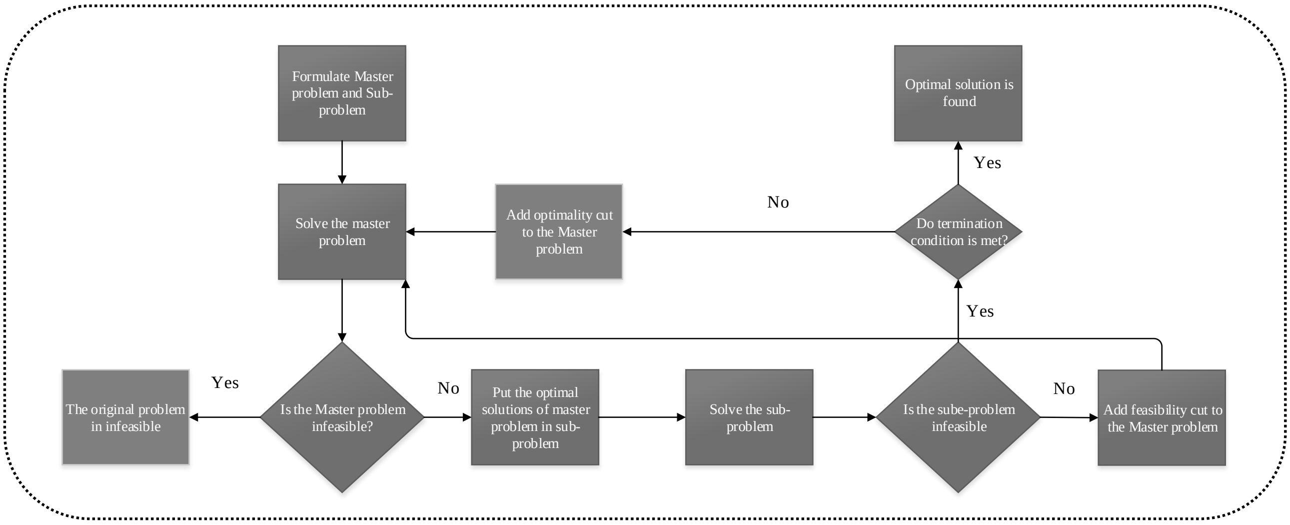

Fig. 1. Flowchart of the solution approach

V. PERFORMANCE EVALUATION

In this section, the performance of the Benders decomposition method is evaluated based on problem instances. This method is implemented in CPLEX 12.6 and runs on a PC with a 2.53 GHz Intel(R) Dual-Core(TM) CPU and 4GB of RAM.

The performance of algorithms is compared with the typical branch and cut method used in CPLEX solver. The branch and the cut algorithm is a combination of two methods of branch and bound and cutting plane, which is widely used to solve integer linear programming. The difference between the branch and the bound method and the branch and cut method is that in the branch and the bound after the relaxed linear problem is solved, and the branching procedure begins when conditions are met. Nevertheless, after the relaxed linear problem is solved in the branch and cut, the cuts are added, and when we cannot add more cut the relaxed problem, then the branching begins.

A set of random data are generated to evaluate the efficiency of the method. Also, other related data is given by the discrete uniform distributions as follows: $P_i \sim U[150;\ 300]$, $q_i \sim U[5;\ 8], t_{ij};\ t_j \sim U[250;\ 600]$ and $d_i \sim U[\frac{\bar{P}+\bar{t}}{2};\ 2 * (\bar{P}+\bar{t})]$. Fixed delivery cost equals to 200, and vehicle capacity is 20. The comparison results of the two methods are reported in Table I and II. To be effective, an upper limit of 300 minutes is considered for all problem instances, and if the problem does not meet the termination conditions, the solution process is interrupted by the user.

In Table I, the proposed method is compared with the solution of CPLEX solver in different sizes of the problem. The number of binary variables, the number of constraints, running time (CPU time), the lower bound of the objective function of the original problem, and the difference between objective functions of two methods are reported in this table. The termination condition for the algorithm is 3 percent GAP, which is calculated as follows:

$$GAP(\%) = \frac{Z_{upper\ bound} - Z_{lower\ bound}}{Z_{lower\ bound}}$$

The Diff column shows the difference between objective functions (lower bounds) of two methods and is calculated as follows:

$$diff(\%) = \frac{\left|Z_{lb}^{Benders} - Z_{lb}^{CPLEX}\right|}{Z_{lb}^{CPLEX}}$$

The number of variables and constraints are calculated based on $(n+1)^2+n+2$ and $2n^2+(n+2)+2$, respectively. In terms of CPU time, the proposed Benders algorithm is faster than CPLEX typical solver. This difference between average running times is 13% in our results. As can be seen, in the last five instances in the branch and cut method, the termination condition of the algorithm has not been met and interrupted by the user in 300 minutes. However, in the case of the proposed method, the last three problem instances do not have been able to meet the termination condition. The percentage of difference in the objective function of the two methods has been reported in the last column of the table. This difference is especially significant in large sizes. Note that this criterion is calculated based on the percentage of relative difference, and the numerical value of this difference is obtained by subtracting the values of the objective function from two methods. The results show that the proposed method is superior to CPLEX solver solutions.

Table I - Comparison of CPLEX solutions and Proposed Benders algorithm

Number of jobs	***Number of variables***	***Number of constraints***	***CPLEX***		***Proposed Benders***		***diff(%)***
			Running time(min)	***Lower bound***	***Running time(min)***	***Lower bound***	
6	85	130	0.41	1765.48	0.38	1765.48	0.00
9	181	274	0.88	2383.40	0.74	2468.44	3.57
12	313	472	1.26	3248.48	1.14	3350.96	3.15
15	481	724	1.49	4276.05	1.27	4413.70	3.22
18	685	1030	2.09	4653.53	1.86	4789.32	2.92
21	925	1390	3.22	5499.47	2.96	5636.36	2.49
24	1201	1804	4.45	6489.97	4.09	6771.68	4.34
27	1513	2272	5.88	7117.16	5.00	7291.32	2.45
30	1861	2794	7.33	8462.07	6.30	8914.80	5.35
34	2381	3574	9.41	10055.89	8.19	10407.52	3.50
38	2965	4450	14.57	11961.91	13.26	12480.72	4.34
42	3613	5422	22.65	13089.02	19.93	13403.52	2.40
46	4325	6490	30.91	14935.96	27.20	15261.28	2.18
50	5101	7654	38.01	16403.23	33.45	16654.80	1.53
55	6161	9244	46.63	17997.41	41.03	18211.60	1.19
60	7321	10984	56.12	20281.17	46.58	20868.50	2.90
65	8581	12874	62.25	21328.66	52.91	21955.35	2.94
70	9941	14914	71.33	25139.74	59.20	25482.20	1.36
75	11401	17104	84.52	26393.93	74.38	26753.10	1.36
80	12961	19444	98.39	28041.85	82.65	28281.80	0.86
90	16381	24574	110.91	29130.42	94.27	29456.36	1.12
100	20201	30304	123.62	30366.26	111.26	30838.40	1.55
110	24421	36634	139.58	34956.50	120.04	35483.30	1.51
120	29041	43564	153.09	38630.80	127.06	39328.91	1.81
140	39481	59224	169.12	44493.59	150.52	45250.73	1.70
160	51521	77284	186.71	49152.01	171.77	49864.40	1.45

Continue Table II - Comparison of CPLEX solutions and Proposed Benders algorithm

Number of jobs	*Number of variables*	*Number of constraints*	*CPLEX*		*Proposed Benders*		*diff(%)*
			Running time(min)	*Lower bound*	*Running time(min)*	*Lower bound*	
180	65161	97744	204.65	51567.16	176.00	52358.11	1.53
200	80401	120604	229.25	53043.06	201.74	54103.60	2.00
250	125501	188254	255.98	58064.19	217.58	59072.67	1.74
300	180601	270904	289.41	129586.14	269.15	132156.35	1.98
350	245701	368554	300.00	176388.09	265.78	179651.03	1.85
400	320801	481204	300.00	221669.33	300.00	225317.61	1.65
450	405901	608854	300.00	265910.24	300.00	269862.92	1.49
500	501001	751504	300.00	325141.56	300.00	329798.66	1.43
Average		121.47		107.46			

In table II, the number of iterations and gap value of solutions from the proposed Benders method are reported. In this case, we use 25 minutes time limit as a termination condition. Hence, any solution that reaches the optimal value at this time will result in a zero value of gap. In comparison with CPLEX solver, the proposed Benders algorithm has been used 33% fewer iterations. Also, the gap of the Benders method for all problem instances is reported in this table. As shown in this table, some initial problems reach optimal solutions. The percentage of the GAP is increased when the size of problems gets larger.

Table III - Results of Proposed Benders algorithm

Proposed Benders					
Number of jobs	*Number of iteration*	*GAP*	*Number of jobs*	*Number of iteration*	*GAP*
6	6	0.00	70	44	0.02
9	7	0.00	75	46	0.03
12	9	0.00	80	48	0.03
15	12	0.00	90	51	0.01
18	13	0.00	100	53	0.03
21	16	0.01	110	55	0.01
24	20	0.02	120	58	0.02
27	21	0.01	140	60	0.03
30	23	0.02	160	62	0.04
34	26	0.02	180	64	0.03
38	28	0.01	200	67	0.03
42	30	0.02	250	69	0.04
46	32	0.03	300	71	0.02
50	35	0.02	350	74	0.03
55	37	0.02	400	76	0.04
60	39	0.01	450	78	0.04
65	42	0.03	500	80	0.03

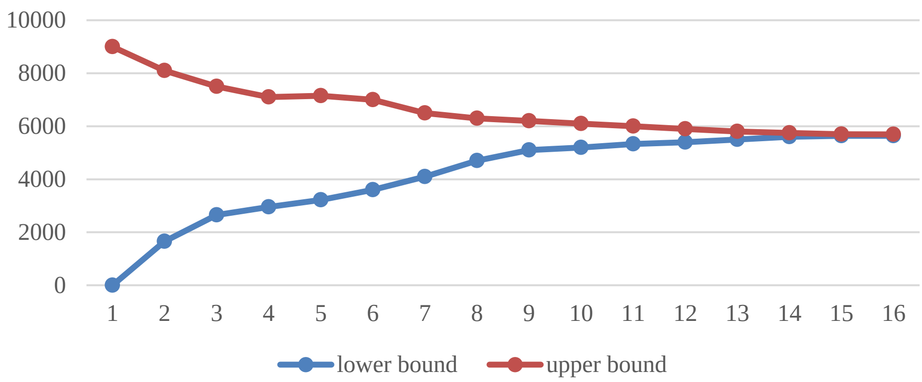

Fig. 2. Convergence of lower bound and upper bound

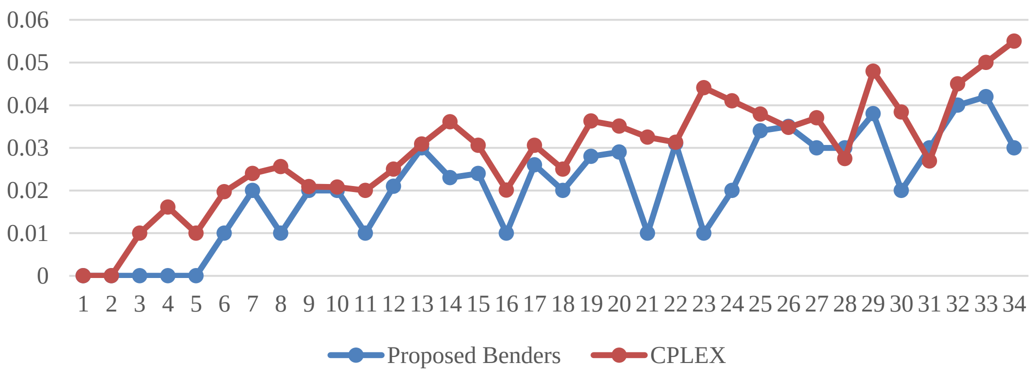

Fig. 3. GAP comparison

In Fig. (2), the convergence of lower bound and upper bound for problem instance n=21 is reported. As can be seen, the GAP between upper and lower bounds is reached to 1 percent in 16 iterations after 30 minutes. Also, in Fig. (3), the GAP between the two methods is compared. When the size of the problem is grown, the GAP value increases for both methods. But the proposed method outperforms the CPLEX method on most of the problems.

In this section, two T-paired tests with a 95% confidence level are performed to illustrate the performance of two algorithms for different instance problems based on the objective function and running times. We conduct this test for problems 6 to 300 jobs. Table II and IV provide results of tests for objective values and running times, respectively. As shown in these tables, p is less than 0.5, which indicates that the mean values of objective functions and running times for two methods are significantly different.

Table III - Estimation for Paired Difference for objective functions

Mean	***StDev***	***SE Mean***	***95% CI for μ_difference***
885	1210	208	(462, 1307)
T-Value	P-Value		
4.26	0.033		

Table IV - Estimation for Paired Difference for Running times

Mean	*StDev*	*SE Mean*	*95% CI for μ_difference*
10.07	10.56	1.93	(6.13, 14.02)
T-Value	P-Value		
5.23	0.007		

VI. CONCLUSION

An integrated production scheduling and distribution problem is described in this paper. Several papers are published in the literature to integrate decisions in the supply chain regarding numerous objectives such as scheduling criteria, cost of operation, customer satisfaction, and so on. In this study, a supply chain with several customers in different locations are considered. Customers place orders at the beginning of the planning horizon and determine their due date for each job. Supply chain management receives these orders and tries to schedule these jobs. After completion of processing jobs, they must be distributed among customers in batch by vehicles. So a processing sequence, assignment of jobs to vehicles, and determining the order of delivery among customers have to be determined. A multi-agent point of view is added to the problem in which two sets of customers (as two agents) have their jobs with a specific objective. At first, a mathematical formulation is developed for this problem. Due to the complexity and structure of the problem and the typical solver's inability, such as CPLEX, a Bender's decomposition approach is presented in solving this problem. It means the problem is decomposed to the master problem and sub-problem. The problem structure is special due to the lack of objective function and the existing integer variable in the sub-problem. Hence, a combinatorial Benders cut is used to deal with this case. The performance of the algorithm is compared with the solution of CPLEX solver in different sizes of the problem. The results show that the proposed algorithm outperforms the branch and cut method used in the CPLEX solver.

There are many research directions for future research. First of all, investigating other scheduling criteria such as makespan or total flow time is considerable. Also, different assumptions can be added to the problem, for example, considering release time or set-up time in scheduling part or limited vehicle and the multi-trip vehicle in the routing part. Other exact approaches or meta-heuristic methods can be the right direction for future research.

REFERENCES

Agnetis, A., Aloulou, M., & Fu, L.L. (2014). Coordination of production and interstage batch delivery with outsourced distribution*European Journal of Operational Research, 238*(1), 130–142.

Agnetis, A., Pascale, G., & Pacciarelli, D. (2009). A Lagrangian approach to single-machine scheduling problems with two competing agents*Journal of Scheduling, 12*(4), 401–415.

Agnetis, A., Mirchandani, P., Pacciarelli, D., & Pacifici, A. (2004). Scheduling problems with two competing agents*Operations research, 52*(2), 229–242.

Armstrong, R., Gao, S., & Lei, L. (2008). A zero-inventory production and distribution problem with a fixed customer sequence*Annals of Operations Research, 159*(1), 395–414.

Baker, K., & Smith, J. (2003). A multiple-criterion model for machine scheduling*Journal of scheduling, 6*(1), 7–16.

Bard, J., & Nananukul, N. (2009). The integrated production–inventory–distribution–routing problem*Journal of Scheduling, 12*(3), 257.

BnnoBRs, J. (1962). Partitioning procedures for solving mixed-variables programming problems '*Numerische mathematik, 4*(1), 238–252.

Chandra, P., Fisher, M., & others (1994). Coordination of production and distribution planning*European journal of operational research, 72*(3), 503–517.

Chen, Z.L., & Vairaktarakis, G. (2005). Integrated scheduling of production and distribution operations*Management Science, 51*(4), 614–628.

Cheng, T., Gordon, V., & Kovalyov, M. (1996). Single machine scheduling with batch deliveries*European Journal of Operational Research, 94*(2), 277–283.

Cheng, T., & Kahlbacher, H. (1993). Scheduling with delivery and earliness penalties*Asia-Pacific Journal of Operational Research, 10*(2), 145–152.

Codato, G., & Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming*Operations Research, 54*(4), 756–766.

Garcia, J., & Lozano, S. (2004). Production and vehicle scheduling for ready-mix operations*Computers & Industrial Engineering, 46*(4), 803–816.

Garcia, J., & Lozano, S. (2005). Production and delivery scheduling problem with time windows*Computers & Industrial Engineering, 48*(4), 733–742.

Geismar, H., Laporte, G., Lei, L., & Sriskandarajah, C. (2008). The integrated production and transportation scheduling problem for a product with a short lifespan*INFORMS Journal on Computing, 20*(1), 21–33.

Hall, N., & Potts, C. (2003). Supply chain scheduling: Batching and delivery*Operations Research, 51*(4), 566–584.

Hall, N., & Potts, C. (2005). The coordination of scheduling and batch deliveries*Annals of operations research, 135*(1), 41–64.

He, Z., Guo, Z., & Wang, J. (2019). Integrated scheduling of production and distribution operations in a global MTO supply chain*Enterprise Information Systems, 13*(4), 490–514.

Herrmann, J., & Lee, C.Y. (1993). On scheduling to minimize earliness-tardiness and batch delivery costs with a common due date*European Journal of Operational Research, 70*(3), 272–288.

Ji, S.f., Peng, X.s., & Luo, R.j. (2019). An integrated model for the production-inventory-distribution problem in the Physical Internet*International Journal of Production Research, 57*(4), 1000–1017.

Joo, C., & Kim, B. (2017). Rule-based meta-heuristics for integrated scheduling of unrelated parallel machines, batches, and heterogeneous delivery trucks*Applied Soft Computing, 53*, 457–476.

Lee, C.Y., & Chen, Z.L. (2001). Machine scheduling with transportation considerations*Journal of scheduling, 4*(1), 3–24.

Lei, L., Liu, S., Ruszczynski, A., & Park, S. (2006). On the integrated production, inventory, and distribution routing problem*IIE Transactions, 38*(11), 955–970.

Li, C.L., & Vairaktarakis, G. (2007). Coordinating production and distribution of jobs with bundling operations*IIE transactions, 39*(2), 203–215.

Li, C.L., Vairaktarakis, G., & Lee, C.Y. (2005). Machine scheduling with deliveries to multiple customer locations*European Journal of Operational Research, 164*(1), 39–51.

Li, K., Zhou, C., Leung, J., & Ma, Y. (2016). Integrated production and delivery with single machine and multiple vehicles*Expert Systems with Applications*, *57*, 12–20.

Lin, W.C., Yin, Y., Cheng, S.R., Cheng, T., Wu, C.H., & Wu, C.C. (2017). Particle swarm optimization and opposite-based particle swarm optimization for two-agent multi-facility customer order scheduling with ready times*Applied Soft Computing*, *52*, 877–884.

Marandi, F., & Fatemi Ghomi, S. (2019). Integrated multi-factory production and distribution scheduling applying vehicle routing approach*International Journal of Production Research, 57*(3), 722–748.

Mula, J., Peidro, D., Diaz-Madronero, M., & Vicens, E. (2010). Mathematical programming models for supply chain production and transport planning*European Journal of Operational Research*, *204*(3), 377–390.

Perez-Gonzalez, P., & Framinan, J. (2014). A common framework and taxonomy for multicriteria scheduling problems with interfering and competing jobs: Multi-agent scheduling problems*European Journal of Operational Research, 235*(1), 1–16.

Pundoor, G., & Chen, Z.L. (2005). Scheduling a production–distribution system to optimize the tradeoff between delivery tardiness and distribution cost*Naval Research Logistics (NRL), 52*(6), 571–589.

Pundoor, G., & Chen, Z.L. (2009). Joint cyclic production and delivery scheduling in a two-stage supply chain*International Journal of Production Economics, 119*(1), 55–74.

Tan, Q., Chen, H.P., Du, B., & Li, X.l. (2011). Two-agent scheduling on a single batch processing machine with non-identical job sizes. In *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)* (pp. 7431–7435).

Rohmer, S., & Billaut, J.C. (2015). Production and outbound distribution scheduling: a two-agent approach. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 135–144).

Selvarajah, E., Steiner, G., & Zhang, R. (2013). Single machine batch scheduling with release times and delivery costs*Journal of Scheduling, 16*(1), 69–79.

Stecke, K., & Zhao, X. (2007). Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode*Manufacturing & Service Operations Management, 9*(2), 206–224.