CrossMark

A coordinated scheduling of delivery and inventory in a multi-location hospital supplied with a central pharmacy

Zakaria Hammoudan[1] · Olivier Grunder[1] · Toufik Boudouh[2] · Abdellah El Moudni[1]

Abstract In today's manufacturing outlook, coordinated scheduling of delivery and inventory represents a leading leverage to enhance the competitiveness of firms which aims to address the new challenge coming from scheduling problems. Though in the last decades this kind of issue has been extensively approached in the literature, a set of constraints and compulsory dispositions strongly increases the complexity of the considered problem. Actors of the pharmaceutical supply chain have to meet various global regulatory requirements while handling, storing and distributing environmentally sensitive products. The studied problem in this paper focuses on a real-case scheduling problem in a multi-location hospital supplied with a central pharmacy. The objective of this work is to find a coordinated production and delivery schedule such that the sum of delivery and inventory costs is minimized. A mixed-integer programming formulation is first detailed to consider the problem under study. Then, a branch-and-bound algorithm is proposed as an exact method and a dedicated heuristic algorithm is highlighted to solve the problem. At last, the experimental results show the efficiency of the proposed solving methods, based on the two following criteria: solution quality and processing time.

1 Introduction and related literature

Today, the expansion of suppliers to accommodate the maximum number of customers is considered as a key factor in the evolution of companies, in order to increase their profits. Industrial companies are continuously assessing their operations with the objective of increasing the overall effectiveness of manufacturing systems. Markets, where these organizations operate, tend to become more complex over time, forcing companies to increase their responsiveness, both in terms of time and cost. The case of the pharmaceutical industry is a good example of how market is driving the change on product development cycles and manufacturing activities. Delivery and inventory scheduling stages are systematically considered to be very difficult functions. They are intended to produce operational plans dealing with several potential conflicting objectives, namely minimizing costs, completion times, and delays or maximizing profit. One important benefit of this coordination is a more efficient management of inventories across the entire supply chain. In traditional inventory management, the optimal production and shipment policies for vendors and customers in a two-echelon supply chain are managed independently. Additionally, these functions are closely related to other areas such as sales, procurement, production execution and control;

✉ Zakaria Hammoudan
   zakaria.hamoudan@utbm.fr

   Olivier Grunder
   olivier.grunder@utbm.fr

   Toufik Boudouh
   toufik.boudouh@utbm.fr

   Abdellah El Moudni
   abdellah.elmoudni@utbm.fr

[1] IRTES-SET, Université de Technologie de Belfort-Montbéliard (UTBM), rue Thierry Mieg, 90010 Belfort Cedex, France

[2] M3M, Université de Technologie de Belfort-Montbéliard (UTBM), rue Thierry Mieg, 90010 Belfort Cedex, France

Springer

Table 1 A numerical example

Jobs	1	2	3.5	4
Hospital's	1	2	1	2
Due date	Thursday 8:00 am	Thursday 8:00 am	Thursday 8:00 am	Thursday 8:00 am
Departure time	Wednesday 06:00 am	Wednesday 06:00 pm	Thursday 2:00 am	Wednesday 06:00 pm
Arrival time	Wednesday 12:00 pm	Wednesday 10:00 pm	Thursday 8:00 am	Wednesday 10:00 pm
Inventory cost	600	200	0	200

hence, they may interface with decisions at the strategic and operational levels. For this reason, the integrated vendor–customer model is developed where the total relevant costs for the customers as well as the vendors have to be minimized. Consequently, determining the production and shipment policies based on an integrated total cost function, rather than several customer's or vendor's individual cost functions, results in the reduction of the inventory costs of the system.

The system under study in this paper is composed of a central pharmacy from which sterilized medical devices have to be delivered before given due dates, to different hospitals located around the central pharmacy. This supply chain process incurs both delivery costs and earliness penalty costs in case the devices are delivered too early.

Therefore, the considered problem is an integrated delivery and inventory problem with due dates constraints, for which we have to minimize the total delivery and holding costs. Therefore, the problem can be formulated from a batch scheduling point of view with a cost objective function or from a lot sizing problem point of view with a time horizon. These two classes of problem have been proven to be equivalent under given conditions [19]. In our case, a batch scheduling approach seems to be more appropriate in the context of the study of the healthcare system with specific constraints for the due dates.

The delivery-inventory problem is denoted as Vendor-Managed Inventory (VMI) problem. The VMI problem is a widely used collaborative inventory management policy in which manufacturers manage the inventory of retailer and take responsibility for making decisions related to the timing and extent of inventory replenishment [7]. VMI partnerships help organizations to reduce demand variability, inventory holding and distribution costs. A pioneering paper is due to Bertazzi et al. [6], where a given set of shipping frequencies is allowed and different products may be shipped at different frequencies. Herer and Levy [14] have considered a system of a central warehouse, a fleet of trucks with a finite capacity, and a set of customers, for each of whom there is an estimated consumption rate, and a known storage capacity. The objective is to determine when to service each customer, as well as the way to be performed by each truck, in order to minimize the total discounted costs. To solve the problem, they have proposed a rolling horizon approach that takes into consideration holding, transportation, fixed ordering and stock out costs. Viswanathan and Mathur [31] have studied a distribution systems with a central warehouse and many retailers that stock a number of different products, where the products are delivered from the warehouse to the retailers by vehicles that combine the deliveries to several retailers into efficient vehicle routes. They have proposed a heuristic that develops a stationary nested joint replenishment policy. These results showed that the proposed heuristic is capable of solving problems involving distribution systems with multiple products. Sindhuchao et al. [29] have considered a system that consists of a set of geographically dispersed suppliers that manufacture one or more non-identical items, and a central warehouse that stocks these items. The warehouse faces a constant and deterministic demand for the items from outside retailers. The items are collected by a fleet of vehicles that are dispatched from the central warehouse. The vehicles are capacitated and must also satisfy a frequency constraint. They studied the case where each vehicle always collects the same set of items. They have formulated and solved the problem by using a branch-and-price algorithm, and then they have proposed a greedy constructive heuristic and a very large-scale neighborhood search algorithm. These results indicate that the constructive heuristic used in conjunction with one of the proposed very large-scale neighborhood algorithms can find near-optimal solutions very efficient. Recently, Archetti et al. [2] have studied a distribution problem in which a product has to be shipped from a supplier to several retailers over a given time horizon. Each retailer defines a maximum inventory level. The supplier monitors the inventory of each retailer and determines its replenishment policy, guaranteeing that no stock out occurs at the retailer (supplier-managed inventory policy). Every time a retailer is visited, the quantity delivered by the supplier is such that the maximum inventory level is reached (deterministic order-up-to level policy). Shipments from the supplier to the retailers are performed by a vehicle of given capacity. They presented a mixed-integer linear programming model, and they derived new additional valid inequalities used to strengthen the linear relaxation of the model. They implemented a branch-and-cut algorithm to solve the model optimally. Then, they have studied two different types of replenishment policies in [3]. The first one is the well-known order-up-to level (OU) policy, where the quantity

shipped to each retailer is such that the level of its inventory reaches the maximum level. The second one is the maximum level (ML) policy, where the quantity shipped to each retailer is such that the inventory is not greater than the maximum level. In this study, Archetti et al. [3] have focused on the ML policy and the design of a hybrid heuristic, and they implemented an exact algorithm for the solution of the problem with one vehicle and designed a hybrid heuristic for the multi-vehicle case. Most recently, Archetti et al. [4], have studied the previous problem with a single vehicle which has a given capacity. The transportation cost is proportional to the distance traveled, whereas the inventory holding cost is proportional to the level of the inventory at the customers and at the supplier. They have proposed a heuristic that combines a tabu search scheme with ad hoc designed mixed-integer programming models. The effectiveness of the heuristic was proved over a set of benchmark instances for which the optimal solution was known.

There are numerous researches on batch scheduling of delivery-inventory problem. Scheduling problems arise in almost any type of industrial production facilities (Pulp and Paper, Metals, Oil and gas, Chemicals, Food and Beverages, Pharmaceuticals, Transportation, Service, Military, etc.) where given operations need to be processed on specified resources. The corresponding scheduling problems are already very difficult to solve [20]. Much research has focused on the same area under various assumptions and objective measures that differ from the considered problem in this paper. Potts [12], Hall[26] and Zhang et al. [35] have studied scheduling problems with non-identical job release times and delivery times, under the assumption that a sufficient number of vehicles is available to deliver the jobs. Kimms [21] has examined the problem of single-machine and proposed two heuristic approaches: randomized regrets based and tabu search approaches. Each production plan is generated without using any information obtained from previous plans. This work has been extended by Kimms [22] with a proposition of a genetic algorithm that dominates the tabu search procedure, both in terms of run-time performance and the ability to find feasible solutions. Pinedo and Michael [24] reviewed different models and solution approaches, and then they explained the complexity of scheduling problems.

Multi-echelon inventory models have attracted much attention, and the integrated approach has been extensively studied. In this way, Grunder [11] considered a single-product batch scheduling problem with the objective of minimizing the sum of production, transportation and inventory cost. Particularly, he assumed that the delivery time depends on the batch sizes and proposes a dynamic programming approach based on a dominance relation property. Wang et al. [33] extended this study with an integrated scheduling problem for single-item supply chain involving due date considerations and an objective of minimizing the total logistics cost. Fu

et al. [10] studied the problem of coordinated scheduling of production and delivery subject to the production window constraints and delivery capacity constraints. They considered both a single delivery time case and multiple delivery time case. Chen [8] reviewed the production and distribution scheduling models and classified these problems in five groups. Problems addressing an objective function that combines machine scheduling with the delivery costs are rather complex. However, they are more practical than those involving just one of the two factors, since these combined-optimization problems are often encountered when real-world supply chain management is considered.

The number of customers and products has been a topic of intense investigation for decades in the integrated supply chain. Although researchers have given a considerable attention on the synchronization of the single-vendor single-customer integrated inventory system, the single-vendor multi-customer integrated inventory case has gotten little attention in regard. Lu [23] developed a one-vendor multi-customer integrated inventory model, while Parija and Sarker [25] extended their published work on single-vendor, single-customer, integrated production-inventory problems with lumpy delivery systems under perfect and imperfect production cycle situations [27]. Lu [23] argued that all the previous studies assumed that the vendor must know the customer's holding and ordering costs, which are quite difficult to estimate unless the customer is willing to reveal the true values. Therefore, he considered another circumstance, in which the objective is to minimize the vendor's total cost per year, subject to the maximum cost that the customer may be prepared to incur. Parija and Sarker [25] introduced the problem of determining the production start time and proposed a method that determines the cycle length and raw material ordering frequency for a long-range planning horizon. The cycle length is restricted to be an integer-multiple of all shipment intervals to the customers as an ideal situation, the solution to which may be sub-optimal. Viswanathan and Piplani [32] proposed a model to study and analyze the benefit of coordinating supply chain inventories by means of common replenishment epochs or time periods. A one-vendor multi-customer supply chain is considered for a single product. Under their strategies, the vendor specifies common replenishment periods and requires all customers to replenish only at pre-determined time periods. However, the authors did not include any inventory cost of the vendor in the model. In most papers dealing with integrated inventory models, the transportation cost is considered only as a part of fixed setup or replenishment cost. Ertogral et al. [9] studied how the results of incorporating transportation cost into the model influence the decision-making process under equal size shipment policies. A fundamental advance in the two-side cost structure is in recognizing how delivery-transportation costs apply to both sides.

Hoque [15] proposed three models for supplying a single-item from a single-vendor to multiple customers under deterministic demand by synchronizing the production flow with equal-sized batch transfer in the first two and unequal-sized batches transfer in the third. In the first two models, all batches forwarded are of exactly the same size but the timing of their shipment is different. In the first of these, the manufacturer transfers a batch to a customer as soon as its processing is finished, whereas in the second a batch is transferred to a customer as soon as the previously sent batch to the customer is finished. In the third model, the subsequent shipment lot sizes increase by the ratio of production rate and sum of demand rates on all the customers. Zavanella and Zanoni [34] proposed a model for a single-vendor multi-customer system, integrated in a shared management of the customers' inventory, so as to pursue a reduction or the stability of the holding costs while descending the chain. Hoque [16] transferred the lot from a vendor to multiple customers with l number of unequal-sized batches first; where the next one is a multiple of the previous one by the ratio ($k > 1$) of the production and the total demand rates, followed by ($n - l$) number of equal-sized batches. The equal-sized batches are restricted to be less than or equal to the lth batch (the largest unequal-sized batch) multiplied by k. The models developed were solved by applying Lagrangian Multiplier method. However, in cases of single-vendor single-customer or single-vendor multi-customer or multi-stage production, synchronization of the production flow by transferring the lot with equal and/or unequal-sized batches was found to lead to the least total cost for some numerical problems. Although Hoque [16] served that purpose, he did not cope with the relaxation of the discussed impractical assumptions. Following this trend of synchronization, Hoque [17] developed two generalized single-vendor multi-customer integrated inventory models by accumulating the inventory at the vendor's and customer's independently, but with the traditional trend of ignoring the cost of benefit sharing. Transportation of each of the batches incurs a transportation cost. In order to implement the models by taking into account the industry reality, he also incorporates them with the relaxation of the discussed impractical assumptions. Battini [5] developed a single-vendor and multi-customer consignment stock inventory model in which many clients can establish a consignment stock inventory policy with the same vendor.

Recently, Jha and Shanker [18] studied an integrated production-inventory model in a single-vendor multi-customer supply chain with lead time reduction under independent normally distributed demand on the customers. They assume a non-identical lead time for the customers and that customers' inventory is reviewed using continuous review policy. Hariga et al. [13] analyzed Hoque's models *I* and *II* studied in Hoque [16], and then they modified some

of Hoque's models. Hariga et al. [13] compared the cost between the results of the models in Hoque [16] and Zavanella and Zanoni [34], and then they concluded that both models are not appropriate as they are using different functional forms of the total setup and ordering costs. Moreover, it is shown that Hoque's model yields impractical solutions for zero transportation costs. When the total setup and ordering cost was adjusted to be similar to the one in Zavanella and Zanoni's model, Hoque's model resulted in a larger total cost.

Existing inventory models for multi-customers are not applicable to pharmaceutical products for several reasons. Pharmaceutical products can be more expensive than other products to purchase and distribute, and shortages and improper use of essential medicines can have a high cost in terms of wasted resources and preventable diseases and death. Therefore, special care should be taken in pharmaceutical inventory decisions to ensure 100 % product availability at the right time, at the right cost, and in good condition to the right customers. The quality of health care industries strongly depends on the availability of pharmaceuticals on time. If a shortage occurs at a hospital, an emergency delivery is necessary, which is very costly and can affect the patient health. Inventory management strategies that are unsuitable for health care industries may lead to large financial losses and a significant impact on patients. Hence, inventory strategies for pharmaceutical products are more critical than those for other products. Thus, a specific inventory model is necessary to control pharmaceutical products, to save patient lives and reduce unnecessary inventory costs.

Here we investigate a delivery-inventory supply chain composed of a central pharmacy which has to deliver pharmaceutical supplies to distant hospitals with a single transporter at given due dates. The objective is to reduce the overall cost which includes the delivery costs and an earliness penalty cost.

The contributions of this paper are twofolds. First, we propose a MIP model to minimize the total delivery and inventory costs for the considered supply chain under the constraints of healthcare systems. Second, we propose an efficient solving algorithm which is compared with two exact methods.

The outline of the remainder of the paper is organized in seven sections. In Sect. 2, the problem definition and formulation is introduced. In Sect. 3, the problem is formulated as a mixed-integer programming (MIP) model. Then, we describe the proposed branch-and-bound algorithm (B&B) as an exact method of resolution in Sect. 4. We develop a heuristic algorithm in Sect. 5 for solving the problem. In Sects. 6 and 7, we eventually provide the experimental results and draw some conclusions and suggest the future research directions.
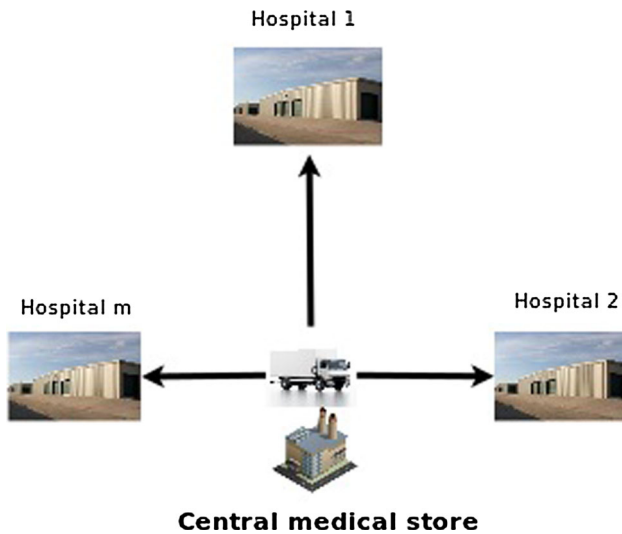
Hospital 1

Hospital m

Hospital 2

Central medical store

Fig. 1 Central pharmacy and multi-location hospital model

2 Problem definition and formulation

We consider a supply chain scheduling problem where there is one central pharmacy which has to deliver medical supplies, or jobs, to m hospital sites, which are the final customers (Fig. 1). Each hospital h orders a finite number of jobs from the central pharmacy.

The following assumptions are considered for this study. First, we will consider a single transporter to deliver the sterilized medical devices as the number of distant hospitals is reduced in practice (less than 4) and the distances with the central pharmacy are quite short. Second, we will only consider direct shipping (i.e., commuter tours), without considering routing considerations between customers [8]. This assumption is explained by the fact that the pharmacy is located in the center of the distant hospitals. Moreover, the road network is centralized on the main town of the central pharmacy; hence, travel times are longer between distant hospitals.

Each round trip between the pharmacy and a hospital h requires a delivery cost η_h as well as a delivery time τ_h. The batches delivered from the central pharmacy to the hospitals can be of different sizes.

The total number of jobs belonging to the same batch cannot exceed the capacity c of the transporter. Each job j has a due date d_j specified by the hospitals and each job has to arrive to the hospital site before its due date. If job j of hospital h arrives before its due date d_j, it will incur as an earliness penalty β_h. Batching and sending several jobs in the batches will reduce the transportation costs.

The objective is to determine the sequence of batches that has to be processed, so that the expected total cost of both central pharmacy and hospitals sites is minimized.

2.1 Notations

The following notations are used in developing the mathematical model:

Parameters

- $J = 1, 2, \ldots, n$: set of all jobs, where n is the total number of jobs,
- $H = 1, 2, \ldots, m$: set of all hospitals,
- j: index for jobs, $j \in J$,
- k: index for batches,
- h: index for hospitals, $h \in H$,
- d_j: due date of job j,
- cl_j: destination of job j, $cl_j \in H$,
- c : capacity of the transporter,
- τ_h: time for the vehicle to deliver a batch to hospital h and to return to the central pharmacy location,
- η_h: delivery cost to deliver a batch to hospital h and to return to the central pharmacy location,
- β_h : hospital earliness penalty function for hospital h.

Primary variables

- $\delta_{jk}^1 = 1$ if the job j belongs to the kth batch, 0 otherwise,
- $\delta_{kh}^2 = 1$ if the batch k belongs to the customer h, 0 otherwise.

Secondary variables

- $y_k = 1$ if the batch k exists and is not empty, 0 otherwise,
- C_j : the arrival time of the job j at the hospital,
- B_k: the arrival time of the batch k at the hospital,
- u_h: number of delivered batches for hospital h.

2.2 Numerical example

To clarify the problem, we consider a simple numerical example in Table (1) as follows. Two hospitals ordered five jobs at the same time (Monday at 8:00 am) and they would receive their products at the same time (Thursday at 8:00 am), that means all the products have the same due date equal to 72 h. The central pharmacy and its hospital customers open 24 h/day. Three jobs ($j = 1, 3$ and 5) for hospital 1 and two jobs ($j = 2$ and 4) for hospital 2. The vehicle capacity is $c = 2$. The transporter delivery cost and time depend on the hospitals' positions with ($\eta_1 = 1000$ Euro, $\tau_1 = 6$ h, and $\eta_2 = 750$ Euro, $\tau_2 = 4$ h) belongs to hospital 1 and 2, respectively, ($\beta_1 = \beta_3 = \beta_5 = 30$ Euro/h and $\beta_2 = \beta_4 = 20$ Euro/h) belongs to hospital 1 and 2, respectively.

The solution is shown in Table (1) for this problem. As it is shown, the vehicle makes three round trips among them two to hospital 1 and one to hospital 2. Three batches

$k_1 = 1$, $k_2 = 2$ and $k_3 = 2$ are denoted. The products arrive at the customers in the batch to which they belong to in the completion time cited in Table (1). The total delivery cost equals $\eta_1 \times 2 + \eta_2 \times 1 = 2750$ Euro and the total storage cost at the hospitals equals $\beta_1 \times [(d_1 - C_1) + (d_3 - C_3) + (d_5 - C_5)] + \beta_2 \times [(d_2 - C_2) + (d_4 - C_4)] = 30 \times [20 + 0 + 0] + 20 \times [10 + 10] = 1000$ Euro. The amount of the objective function is 3750 Euro.

3 The mixed-integer programming model

The pharmaceutical supply chain has many aspects that need to be considered in a supply chain model. However, by taking all concerned factors into account, the model would be of so high complexity that it would be extremely hard for analysis. In this section, the mathematical programming model of the above-mentioned problem is presented. Using the structural properties, we develop a MIP model for the mentioned problem as follows:

$$Min\, Z = \sum_{h=1}^{m} \eta_h u_h + \sum_{j=1}^{n} \beta_{cl_j}(d_j - C_j) \qquad (1)$$

Subject to :

$$\sum_{k=1}^{n} \delta_{jk}^{1} = 1, \quad j = 1, \ldots, n \qquad (2)$$

$$\sum_{h=1}^{m} \delta_{kh}^{2} \leq 1, \quad k = 1, \ldots, n \qquad (3)$$

$$\delta_{k,cl_j}^{2} \geq \delta_{jk}^{1}, \quad j, k = 1, \ldots, n \quad \text{and } k \leq j \qquad (4)$$

$$u_h = \sum_{k=1}^{n} \delta_{kh}^{2}, \quad h = 1, \ldots, m \qquad (5)$$

$$y_k \geq y_{k+1}, \quad k = 1, \ldots, n - 1 \qquad (6)$$

$$\sum_{j=k}^{n} \delta_{jk}^{1} \leq c, \quad k = 1, \ldots, n \qquad (7)$$

$$C_j \leq d_j, \quad j = 1, \ldots, n \qquad (8)$$

$$B_{k+1} \geq B_k, \quad k = 1, \ldots, n - 1 \qquad (9)$$

$$B_{k+1} - B_k \geq \sum_{h=1}^{m} \tau_h(\delta_{k+1,h}^{2} + \delta_{kh}^{2}), \qquad (10)$$
$$k = 1, \ldots, n - 1 \ and \ h \in H$$

$$C_j = \sum_{k=1}^{n} B_k \times \delta_{jk}^{1}, \quad j = 1, \ldots, n \qquad (11)$$

$$C_j \geq 0, \quad j = 1, 2, \ldots, n \qquad (12)$$

$$\delta_{jk}^{1}, \delta_{kh}^{2} \in \{0, 1\}, \quad j, k = 1, \ldots, n \quad \text{and} \quad h = 1, \ldots, m \qquad (13)$$

The objective function (1) minimizes the sum of the delivery costs, through the $\eta_h u_h$ term, and the customers earliness penalty, through $\beta_{cl_j}(d_j - C_j)$. Constraint (2) guarantees that each job must be scheduled exactly in one batch. In this constraint, the jobs will be batched only in the batch which it belongs to. Constraints (3 and 4) force each batch to be delivered to the customer it belongs to. Constraint (5) calculates the number of batches delivered to each customer. Constraint (6) guarantees that no empty batch is allowed. Constraint (7) prevents the number of jobs scheduled in one delivery batch to exceed the capacity of the vehicle. Constraint (8) indicates that arrival time of each job is at least equal to the contracted due date for each customer. Constraint (9) orders the batches in the increasing order of their arrival times. Constraint (10) expresses the minimum interval duration between the arrivals of two consecutive batches has to be greater than the delivery time of the transporter. Constraint (11) represents the relation between the completion time of the jobs and the arrival time of the batch they belong to. This constraint is represented in a nonlinear way in this mathematical representation to facilitate the understanding of the problem. Constraints (12) and (13) define the range of the variables.

For ease of reference, we denote this problem: Multiple Customers Batching Delivery Scheduling Problem (MCBDSP).

The complexity of the MCBDSP is still an open question. To the best of our knowledge, no polynomial algorithm can solve this problem. However, from simulation experiments, we observe that the problem is still intractable on an empirical basis. In the next section, a B&B with a lower bound is described to solve the problem as an exact method.

4 Branch-and-bound algorithm

In this section, we describe the B&B algorithm that we have developed to solve the MCBDSP. The objective of this B&B is to solve small to medium-sized instances, and to be a reference for validating the efficiency of the proposed heuristic algorithm. This B&B algorithm maintains a list of subproblems (nodes) whose union of feasible solutions contains all feasible solutions of the original problem. The list is initialized with the original problem itself. In each major iteration, the algorithm selects a current subproblem from the list of unevaluated nodes. This branching seems to be natural; however, the number of branches will be very large for large problems. Consequently, if this method is used in the B&B algorithm, it may take too much time to find optimal solutions, as redundant schedules

would be checked repeatedly. Yet, several of the subproblems would already have been eliminated upon the generation of nodes, since the search tree includes redundant solutions.

At each node of the search tree, the number of products that still need to be delivered to each customer has to be updated. Iterations are performed until the list of subproblems to be processed is empty. The crucial part of a successful B&B algorithm is the computation of the lower bounds. Therefore, we have developed a lower bound described in the next part.

Efficient lower bound would significantly reduce the time and efforts needed for the B&B method. Based on the main feature of the problem, the lower bound value for the problem is the summation of lower bounds on the total earliness cost and the transportation cost. We assume that w is a partial batch sequence solution, $z(w)$ is the evaluation of w, and $r_h(w)$ is the number of products remaining at the customer's h for partial solution w. This notation will be used throughout this part.

In each node, the solutions are built from the last batch to the first one and the evaluation of the partial or complete solution is processed with backward equations. The research of a solution starts by constructing a partial solution w. Then, the remainder of products is added in order to get a complete solution, with the objective of achieving a minimum delivery cost. Therefore, more the transporter will be loaded, more this lower bound will be efficient.

Proposition 1 *For a partial solution w, a lower bound for the delivery cost of the remaining products is given by:*

$$\sum_{h=1}^{m} \left\lceil \frac{r_h}{c} \right\rceil \times \eta_h \tag{14}$$

Proof For each customer h, if $r_h(w)$ is the number of products remaining to be delivered, the number of round trips will be equal to $\left\lceil \frac{r_h}{c} \right\rceil$, and the delivery cost of the remaining products is as denoted in Eq. (14). □

We add the partial solution w to the solution found in equation (14) to get the lower bound of the current node under study.

Corollary 1 *The lower bound $LB(w)$ of the partial solution w is given as follows:*

$$LB(w) = z(w) + \sum_{h=1}^{m} \left\lceil \frac{r_h}{c} \right\rceil \times \eta_h \tag{15}$$

Proof Straightforward. □

The mathematical model and the B&B algorithm developed in the previous sections could solve small to medium-sized instances; however, the time of resolution to solve large-sized instances grows exponentially in the experimental results. Therefore, developing fast heuristic algorithm to yield near-optimal solutions in a reasonable running time is still of great importance. In the next section, a solving method is proposed to solve the problem.

5 Heuristic algorithm

In this section, a heuristic algorithm, which is denoted Batching and Scheduling algorithm (B&S), is proposed. This algorithm is composed of two steps, the first one consists in defining the size of the batches and the second one will schedule them according to the different constraints of the problem.

The B&S algorithm starts by generating an initial solution through the means of a progressive constructive procedure. Then, the above-mentioned two-steps process is applied until a predefined stop condition is satisfied. At first, some elements of the current solution are constructed. Then, a local improvement phase based on a swap operator is applied to the reconstructed solution in order to improve its quality. Finally, B&S chooses the optimum solution between the current solution and the solution obtained from the improvement procedure.

Let us denote that $(q_k; cl_k; B_k)$ is the notation which will be used for a solution of a batch k, where the first term q_k describes the number of jobs in this batch, the second term cl_k describes the customers destination of batch k and the third term B_k is the arrival time of this batch. For example, a solution of three batches, which contains 2, 3 and 2 jobs, respectively, belongs to customers 2, 3 and 1, respectively, and arrives at due dates 1000, 1015 and 1020, respectively, will be written as follows:

$$[(2_1; cl_2; 1000), (3_2; cl_3; 1015), (2_3; cl_1; 1020)]$$

Based on the prune rule, the following heuristic algorithm is proposed as follows: for level 0, there is no job. For the first level, which includes only last job n, there is only one possible joint solution which is $(1; cl_n; B_n)$. For level k (includes q jobs), all "good" solutions for a number of k jobs will be kept. The process to build "good" solutions for level k is described as follows: (1) build solutions of level k by considering all the solutions in the retained "good" solutions of all the previous levels from 1 to $(k-1)$. For each retained solution of level $k' \le k$, a new solution of level k is built by simply adding a batch of $(k-k')$ jobs, if this is possible. Then, this procedure is repeated until the level n is reached.

The details of the algorithm (1) are presented as follows: The generation of the initial solution and the construction procedure is represented from line 1 to 4. Then, the batch sizing procedure is represented from line 5 to 17 according

to a scattering/gathering procedure. The improvement procedure is called in line 14, and then it is described in Algorithm 2.

The batch sizing procedure, performed in an iterative way, extends a partial solution by adding one job from a set J of all jobs. The construction of the good solution advances progressively and in a hierarchical manner. The process starts from the last job and arrives recursively to the first one. The jobs are distributed to the customers to whom they belong, and the batches sizes are defined according to a scattering/gathering procedure described in Algorithm 1.

Algorithm 1 The Batching & Scheduling Algorithm

```
1  for j ← 0 to n do
2      currentJob ← n − j + 1 ;
3      jobDest ← cl[CurrentJob] ;
4      nbrPartsDelivered [jobDest]++ ;
5      for level ← 0 to j do
6          if (TestOfCapacity()) then
7              currentSol ← GetBestSol(level) ;
8                  for h ← 1 to m do
9                      if (TestQuantityDeliv()) then
10                         AddBatch(currentSol) ;
11                     end if
12                 end for
13                 Evaluate(currentSol) ;
14                 Improvement(currentSol) ;
15                 CompareAndUpdate(currentSol) ;
16         end if
17     end for
18 end for
19 bestSolution ← GetBestSol n ;
```

In this algorithm, the number of delivered jobs j varies from 0 to n (line 1). For each level of j delivered jobs, the different partial solutions are built from the solution of previous levels ($<j$). Moreover, the necessary number of batches to these solutions is added, to complete the partial solution of level j. After every product addition to level j, the partial solution of this level is completed by adding the necessary delivery scheme to the considered solution in the list of all kept solutions from 0 to $(j − 1)$, to obtain the new list of solutions of level j. A test of verification of the capacity of the transporter used is done directly after each advancement in level (See line 6 in Algorithm 1).

The final step of each level j is denoted in line 15, which is mentioned in Algorithm 1. In this phase, the good solution is memorized and inscribed to level j. After that, a new level $(j + 1, j + 2 ...)$ is started till reaching level n.

In the improvement phase, all consecutive batches are swapped, by starting from the last batch recursively to the first one, while the index of the batch is positive (See line 5 in Algorithm 2). After every swap operation, the new solution is kept if it is better than the current solution. If

not, a new swap operation is generated. The improvement operation stops when the index of batches equals 0.

Algorithm 2 Improvement phase of the Batching & Scheduling Algorithm

---
```
1  Input a solution currentSol
2  bestEval ← Evaluate(CurrentSol) ;
3  indexBatch ← NumberOfBatches -1 ;
4  while (indexBatch ≥ 0 ) do
5      Swap(currentSol, indexBatch , indexBatch +1) ;
6      if (Evaluate (currentSol) < bestEval)  7 then
8          bestEval ← Evaluate(currentSol) ;
9          improve ← true ;
10     else
11         Swap(currentSol , indexBatch, indexBatch +1) ;
12         Evaluate(currentSol) ;
13     end if
14     indexBatch ← indexBatch - 1 ;
15 end while
```

Let's take an example to explain the application of the B&S in Algorithm 1, to illustrate the MCBDSP. We consider a problem of three jobs and two customers. The due dates associated with these jobs equal 1000, 1100, 1150, where jobs 1 and 3 have due dates 1000 and 1150 and belong to customer 1 and job 2 has due date 1100 and belongs to customer 2. The transport delivery cost and time depend upon the customer's location with ($\eta_1 = 20, \tau_1 = 60 \, u.t$ and $\eta_2 = 15, \tau_2 = 40 \, u.t$) belonging to customer 1 and 2, respectively. The customers' holding costs are defined as follows: ($\beta_{cl_1} = \beta_{cl_3} = 30$ and $\beta_{cl_2} = 15$), belonging to customer 1 and 2, respectively. The B&S process is described in detail as follows: the process starts by the last job recursively to arrive to the first one.

1. For $j = 1$; currentJob = 3, there is only one possible joint solution which is $(1; cl_1; 1150)$.

2. For $j = 2$; currentJob $j = 3$ or 2, there are different possible solutions. Firstly, a complementary solution is built by simply adding a batch of $(2 − 1)$ job to the previous delivery solution. The potential delivery scheme is equal to 2 batches:

   $[(1; cl_1; 1150), (1; cl_2; 1100)]$

   Due to the improvement phase, two solutions could be obtained according to the swap operation which are:

   $[(1; cl_2; 1100), (1; cl_1; 1150)]$ and
   $[(1; cl_1; 1150), (1; cl_2; 1100)]$

   Then the two potential joint solutions are compared and the good solution is kept, which is:

   $[(1; cl_2; 1100), (1; cl_1; 1150)]$

3. For $j = 3$. currentJob $j = 3, 2$ or 1, based on the delivery scheme of the first step, the new delivery schemes are:

   $[(1; cl_1; 1150), (1; cl_2; 1100), (1; cl_1; 1000)]$ and
   $[(1; cl_1; 1150), (1; cl_1; 1000), (1; cl_2; 1100)]$.

Due to the improvement phase, two solutions could be obtained according to the swap operation which are:

$$[(1; cl_2; 1100), (1; cl_1; 1000), (1; cl_1; 1150)] \text{ and}$$
$$[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$$

Then the four potential joint solutions are compared and the good solution is kept, which is:

$$[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$$

After that, based on the delivery scheme of the second step, the delivery scheme on this step is:

$$[(1; cl_2; 1100), (1; cl_1; 1150), (1; cl_1; 1000)]$$

In the improvement phase, a new solution could be obtained which is:

$$[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$$

The two potential joint solutions are compared and the good solution is kept, which is:

$$[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$$

Finally, the two potential joint solutions kept in each level are compared and the best final solution is recovered, which is:

$$[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$$

6 Experimental results

In this section, a set of problems taken from the central pharmacy data with different sizes are used for this study. The computational experiments are carried out to test the performance of the three techniques of resolution used to solve the problem under study: the MIP model solved by CPLEX, the proposed B&B algorithm and the developed B&S heuristic algorithm.

The performance of B&S was measured by the average error gap compared to the fast exact method (which is the developed B&B algorithm in this study) and was defined as ER(B&S/B&B)=$(E_{B\&S}-E_{B\&B})/E_{B\&B}$ where $E_{B\&S}$ denotes the best evaluation found by the heuristic algorithm and $E_{B\&B}$ the best evaluation of the branch-and-bound algorithm. The performance of the proposed B&B procedure was measured by its Central processing unit (Cpu) time needed to find the optimal solutions and was compared with the CPLEX solver that solves the MIP model directly. Both the B&B procedure and the B&S were programmed in JAVA language and implemented through a desktop Intel core 2 processor operating at 2.67 GHz clock speed and 4 GB RAM. The MIP model was solved by CPLEX on the same machine.

The maximum solving time allowed for these instances is 1 h. The reference of time limit of resolution is based on the real time of the preparation of a schedule in the actual case. As a comparison, CPLEX solver is used to exactly solve the model with small scale random instances. Some adjustments are done on the parameters of research in CPLEX order to accelerate the research of solutions.

According to the confidentiality of the data base of the central pharmacy under study, several cases of problem were considered for which several instances were generated randomly.

6.1 Test cases

The characteristics of orders to schedule differ by customers, transporter capacity, quantity delivered, due date, transporter time, transporter cost and the storage cost at each customer. Three cases are considered to test the proposed methods. The characteristics of the case are listed in Table 2. For each case {A, B and C}, the number of products n, the number of customers h, the transporter time τ_h, the transporter cost (η_h), and the storage cost at each customer (β_h) are displayed.

In the first case, η_h is higher than β_h, where η_h and β_h are randomly generated from the uniform distribution with ranges [1000, 1500] and [1, 5], respectively. In the second case, η_h is generated in the same way and with the same distribution with ranges [1000, 1500] and β_h of the first case is multiplied by 10, where β_h is randomly generated from the uniform distribution with ranges [10, 50]. In the third case, β_h is calculated by multiplying the ranges of the first case by 100, where β_h is randomly generated from the uniform distribution with ranges [100, 500].

Table 2 Main characteristics of the test cases

Case	n	h	β_h(Euro)	η_h(Euro)	τ_h(Hours)
A	10	2, 3, 4	[1; 5]	[1000; 1500]	[3; 5]
	20	2, 3, 4			
	30	2, 3, 4			
	40	2, 3, 4			
B	10	2, 3, 4	[10; 50]	[1000; 1500]	[3; 5]
	20	2, 3, 4			
	30	2, 3, 4			
	40	2, 3, 4			
C	10	2, 3, 4	[100; 500]	[1000; 1500]	[3; 5]
	20	2, 3, 4			
	30	2, 3, 4			
	40	2, 3, 4			

6.2 Comparison of the performance of the B&B algorithm and the MIP model

Both methods of resolution, B&B algorithm and MIP model solved by CPLEX, find optimal solutions. Their performances are measured by their Cpu time, then the fast method will be compared to the developed heuristic algorithm B&S .

The CPLEX solver, which is used to solve the MIP model, finds the optimal solution. However, its computational time grows exponentially as the instance size increases, regardless of the parameters of the studied problem. In contrast, the proposed B&B algorithm is influenced by the value of the parameters used and the increase of the complexity of the problem. With small to medium-sized instances, the computational time of the proposed B&B algorithm will never exceed one hour. The results show that the B&B algorithm runs much faster than the CPLEX solver.

For the problem of the class A in Table 3, we notice that B&B which is supported by the lower bound runs faster than the CPLEX solver. The CPLEX solver finds the optimal solution but its computational time grows rapidly as the size of the instance and the number of customers increase. Conversely, the computational time of the B&B algorithm is very short, which explains the efficiency of the lower bound used in the B&B method to give the optimal solution from small to medium-sized instances. In this case, the two methods solve the problem rapidly. In these experiments, the optimal solution corresponds to fully loaded batches. In this case, the total holding cost is less than the total transporter cost. Consequently, this configuration is the least complex to solve, because the batches have to be fully loaded in order to minimize the delivery cost.

In the second class of the problem B in Table 4, the problem becomes harder to solve with CPLEX onset from 30 products regardless of the number of transporters. The B&B algorithm runs faster than the CPLEX solver, but the gap between the two methods becomes significantly prominent as the number of products and transporters increases. In this case, the time of resolution of the CPLEX solver starts to increase rapidly according to the variation of β in [10, 50]. In the optimal solution we noticed that the number of batches is increased and the number of products by batch is decreased gradually.

In the third class of the problem C in Table 5, we observe that the B&B algorithm runs much faster than the CPLEX solver when the number of products is more than 10 products. Interestingly, the efficiency of CPLEX decreases drastically, where the MIP model solves only the instances of ten products with 2, 3 and 4 customers. In this case, the computational time of the two methods becomes very large so that the variation of β equals [100, 500], where in the optimal solution the batches are very lightly loaded, but the proposed B&B algorithm is still more efficient than the results of the MIP model.

These results show the efficiency of the proposed B&B method to give the optimal solution from small to medium-sized instances. In the next section, the performance of the B&B algorithm will be compared to the proposed heuristic algorithm B&S.

6.3 Comparison of the quality of solutions

In this section, the performance of the proposed B&S heuristic algorithm is analyzed thoroughly by comparing these results with the performance of the proposed exact methods.

The three considered cases are found in Table 6. For each case, three scenarios are considered beginning with three hospitals in use, then four and five. Moreover, for each case, the number of products sets as 10, 20, 30 and 40, respectively. In each case, the customer storage cost β_h is generated from a discrete uniform distribution in the interval [1, 5], [10, 50], and [100, 500] Euro for the three cases, respectively.

In the computational study, the following parameters are used: the vehicle's capacity is randomly generated from the uniform distribution with range $[n/5, 2n/5]$; further, its round-trip delivery time for each customer is randomly generated from the uniform distribution with range [3, 5] h. The due dates $(d_j)_{j=1...n}$ are uniformly separated with values randomly generated.

Considering the different parameters, 36 situations of the problem are tested. For each situation, 25 problem instances are generated to study the performance of the B&S. Based on the results of the exact methods, the error

Table 3 Computational results of instances with $\beta = [1, 5]$	Class of problem↓	Size(n)→	10		20		30	
		(h)↓	CPLEX CpuT (s)	B&B CpuT (s)	CPLEX CpuT (s)	B&B CpuT (s)	CPLEX CpuT (s)	B&B CpuT (s)
		2	0.64	0.006	53.67	0.046	144.842	0.221
	Case A	3	0.505	0.005	45.31	0.385	197.274	1.749
		4	1.660	0.019	43.25	0.891	288.585	308.997

Table 4 Computational results of instances with $\beta = [10, 50]$

Class of problem↓	Size(n)→	10		20		30	
	(h)↓	CPLEX CpuT (s)	B&B CpuT (s)	CPLEX CpuT (s)	B&B CpuT (s)	CPLEX CpuT (s)	B&B CpuT (s)
	2	3.439	0.006	462.669	0.149	>3600	1.471
Case B	3	2.106	0.006	515.522	1.133	>3600	63.433
	4	4.753	0.023	474.654	6.161	>3600	>3600

Table 5 Computational results of instances with $\beta = [100, 500]$

Class of problem↓	Size(n)→	10		20		30	
	(h)↓	CPLEX CpuT (s)	B&B CpuT (s)	CPLEX CpuT (s)	B&B CpuT (s)	CPLEX CpuT (s)	B&B CpuT (s)
	2	15.208	0.008	>3600	0.301	>3600	6.602
Case C	3	18.749	0.008	>3600	2.882	>3600	241.455
	4	29.442	0.020	>3600	32.497	>3600	>3600

Table 6 The error ratio results for $\eta_h \in [1000, 1500]$ Euro and $\beta_h \in [1, 5], [10, 50], [100, 500]$ Euro for three cases, respectively

Class of problem↓		The error ratio		
		ER = $(E_{B\&S}-E_{B\&B})/E_{B\&B}$		
	(h)→ (n)↓	3	4	5
Case A	10	9.11 %	2.17 %	0.21 %
	20	7.13 %	7.67 %	0.35 %
	30	9.52 %	2.32 %	0.83 %
	40	8.80 %	5.27 %	7.99 %
Case B	10	0.00 %	1.15 %	0.04 %
	20	8.62 %	3.68 %	0.14 %
	30	8.19 %	4.03 %	0.35 %
	40	12.43 %	12.28 %	4.60 %
Case C	10	5.90 %	2.18 %	0.98 %
	20	5.10 %	2.89 %	1.59 %
	30	9.00 %	12.19 %	2.95 %
	40	12.12 %	10.19 %	7.58 %
	Overall average = 5.82 %			

ratio is defined as ER(B&S/B&B) = $(E_{B\&S} - E_{B\&B})/E_{B\&B}$, where $E_{B\&S}$ denotes the mean evaluation of the solution generated by the proposed B&S, and $E_{B\&B}$ denotes the mean evaluation of the solution generated by exact methods. The results are displayed in Table 6.

Table 6 shows clearly that the overall average equals 5.82 % which demonstrates that the proposed B&S is capable of generating near-optimal solutions within a reasonable amount of Cpu time. One of the reasons may be the improvement phase which is presented in the heuristic algorithm (Sect. 5). In each case, we observe that the average error ratio appears in an increasing trend as the value of n increases.

6.4 Comparison of the computational time of solving methods

Tables (7, 8, 9) show the solution time obtained for each method. In the computational study, the following parameters are used: the vehicle's capacity is randomly generated from the uniform distribution with range [$n/5$, $2n / 5$]; further, its round-trip delivery time for each hospital is randomly generated from the uniform distribution with range [3, 5] h. The due dates $(d_j)_{j=1...n}$ are uniformly separated with values randomly generated.

Moreover, for each case, the number of jobs set as 10, 20, 30 and 40, and the number of hospitals as 2, 3 and 4 for each case.

The parameters are generated with a magnitude order which is consistent with those of the central pharmacy. For each combination, 25 problem instances are randomly generated and the average Cpu time for each resolution method is collected.

The results show that the heuristic algorithm runs much faster than the B&B algorithm. In this class, the resolution of the B&B algorithm is acceptable, which explains the efficiency of the lower bound used in the B&B algorithm to give the optimal solution for small to medium-sized instances. The computational time of the proposed B&S will never exceed 0.3 s; moreover, the B&S can give optimal or near-optimal solutions for all of the situations.

In Table 7, the results show that it was possible to solve all the instances with the three proposed methods. In this case, the total storage cost at the customer's $\sum_{j=1}^{n} \beta_{cl_j}(d_j - C_j)$, which constitutes the second part of the objective function (1), will be less than those of the total transporter cost $\sum_{h=1}^{m} \eta_h u_h$, which constitutes the first part of the objective function (1). This configuration is the least complex to solve, because the vehicle is fully loaded

Table 7 Computational results for $\eta_h \in [1000, 1500]$ Euro and $\beta_h \in [1, 5]$ Euro

Class of problem→		Class A		
	$(h)\rightarrow$ $(n)\downarrow$	2 CpuT (s)	3 CpuT (s)	4 CpuT (s)
B&B	10	0.006	0.005	0.019
B&S		0.015	0.013	0.012
B&B	20	0.046	0.385	0.891
B&S		0.026	0.031	0.041
B&B	30	0.221	1.749	20.934
B&S		0.044	0.064	0.103
B&B	40	1.785	40.361	308.997
B&S		0.069	0.121	0.193

Table 8 Computational results for $\eta_h \in [1000, 1500]$ Euro and $\beta_h \in [10, 50]$ Euro

Class of problem→		Class B		
	$(h)\rightarrow$ $(n)\downarrow$	2 CpuT (s)	3 CpuT (s)	4 CpuT (s)
B&B	10	0.006	0.006	0.023
B&S		0.015	0.013	0.012
B&B	20	0.149	1.133	6.161
B&S		0.025	0.031	0.042
B&B	30	1.471	63.433	>3600
B&S		0.045	0.061	0.101
B&B	40	24.229	>3600	>3600
B&S		0.068	0.110	0.186

Table 9 Computational results for $\eta_h \in [1000, 1500]$ Euro and $\beta_h \in [100, 500]$ Euro

Class of problem→		Class C		
	$(h)\rightarrow$ $(n)\downarrow$	2 CpuT (s)	3 CpuT (s)	4 CpuT (s)
B&B	10	0.008	0.008	0.020
B&S		0.014	0.013	0.013
B&B	20	0.301	2.882	32.497
B&S		0.026	0.031	0.036
B&B	30	6.602	241.455	>3600
B&S		0.046	0.069	0.110
B&B	40	78.240	>3600	>3600
B&S		0.074	0.135	0.213

according to the cheapness of the storage cost at the hospital.

In Table 8, the problem becomes harder to solve for the proposed B&B algorithm. In this case, the time of resolution of the B&B algorithm exceeds the proposed time limit onset from four hospitals if the number of products equals 30, and onset from three hospitals when the number of products equals 40. Here, the number of batches is increased and the number of products by batch is decreased.

In the third class of problem, in Table 9, the B&B algorithm solves instances until 30 products with four hospitals. Its processing time grows progressively when the number of hospitals and products increase. In this case, the vehicle is very lightly loaded.

These results show that the proposed B&B algorithm is efficient for small to medium-sized instances and finds optimal solutions, and the B&S proposed algorithm gives an optimal or a near-optimal solutions for small to large-sized instances.

7 Conclusion

In this paper a real-life delivery and inventory problem from the pharmaceutical industry is addressed. A central pharmacy delivers products to multiple heterogeneous hospitals sites with a single transporter. The transporter serves every hospital separately. It is supposed that each job that arrives in the hospital before its due date will incur an earliness penalty cost. The objective is to minimize the total cost defined by the weighted sum of the delivery cost and the earliness cost.

Firstly, we focused on the development of a complete deterministic model formulated as a mixed-integer programming model. Then, in a subsequent step, a branch and bound based on a lower bound is developed. Secondly, we described an effective heuristic algorithm based on the determination of the batch sizing and the batch scheduling of the problem. The efficiency of the proposed heuristic algorithm guarantees the determination of a feasible schedule for any given set of requests of the central pharmacy.

The proposed heuristic algorithm is compared with the proposed exact methods. The results illustrate the interesting potential of the proposed approach. The branch and bound proved to be very efficient. Indeed, it proved to be far more efficient than the existing MIP model for solving the problem. The efficiency of the branch-and-bound algorithm is attributable to the tightness of the lower bounds derived. Moreover, efficiency of branch and bound increases for problem instances with a medium number of products. A very effective heuristic algorithm procedure is developed. The results show clearly that the proposed heuristic algorithm is capable of generating near-optimal solutions within a reasonable Cpu time.

There are several directions for future research. Firstly, the model could be advanced by allowing the vehicle routing with integrated delivery and storage cost. Secondly,

the setup time and cost could be integrated into the production stage, and the volume of products into the delivery stage. Finally, we aim to extend the considered model to the multi-transporters case, where each transporter could be assigned to one customer.

References

1. Aptel O, Pourjalali H (2001) Improving activities and decreasing costs of logistics in hospitals: a comparison of US and French hospitals. Int J Account 36(1):65–90
2. Archetti C, Bertazzi L, Laporte G, Speranza MG (2007) A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transp Sci 41(3):382–391
3. Archetti C, Bertazzi L, Paletta G, Speranza MG (2011) Analysis of the maximum level policy in a production–distribution system. Comput Oper Res 38(12):1731–1746
4. Archetti C, Bertazzi L, Hertz A, Speranza MG (2012) A hybrid heuristic for an inventory routing problem. INFORMS J Comput 24(1):101–116
5. Battini D, Gunasekaran A, Faccio M, Persona A, Sgarbossa F (2010) Consignment stock inventory model in an integrated supply chain. Int J Prod Res 48(2):477–500
6. Bertazzi L, Speranza MG, Ukovich W (1997) Minimization of logistic costs with given frequencies. Transp Res B Methodol 31(4):327–340
7. Borade AB, Sweeney E (2015) Decision support system for vendor managed inventory supply chain: a case study. Int J Prod Res 53(16):4789–4818
8. Chen Z-L (2010) Integrated production and outbound distribution scheduling: review and extensions. Oper Res 58(1):130–148
9. Ertogral K, Darwish M, Ben-Daya M (2007) Production and shipment lot sizing in a vendor–buyer supply chain with transportation cost. Eur J Oper Res 176(3):1592–1606
10. Fu B, Huo Y, Zhao H (2010) Coordinated scheduling of production and delivery with production window and delivery capacity constraints. In: Chen B (ed) Algorithmic aspects in information and management. Springer, New York, pp 141–149
11. Grunder O (2010) Lot sizing, delivery and scheduling of identical jobs in a single-stage supply chain. Int J Innov Comput Inf Control 6(8):3657–3668
12. Hall LA, Shmoys DB (1992) Jackson's rule for single-machine scheduling: making a good heuristic better. Math Oper Res 17(1):22–35
13. Hariga M, Hassini E, Ben-Daya M (2013) A note on generalized single-vendor multi-buyer integrated inventory supply chain models with better synchronization. Int J Prod Econ 154:313–316
14. Herer YT, Levy R (1997) The metered inventory routing problem, an integrative heuristic algorithm. Int J Prod Econ 51(1):69–81
15. Hoque M (2008) Synchronization in the single-manufacturer multi-buyer integrated inventory supply chain. Eur J Oper Res 188(3):811–825
16. Hoque M (2011a) Generalized single-vendor multi-buyer integrated inventory supply chain models with a better synchronization. Int J Prod Econ 131(2):463–472
17. Hoque M (2011b) An optimal solution technique to the single-vendor multi-buyer integrated inventory supply chain by incorporating some realistic factors. Eur J Oper Res 215(1):80–88
18. Jha J, Shanker K (2012) Single-vendor multi-buyer integrated production-inventory model with controllable lead time and service level constraints. Appl Math Model 37(4):1753–1767
19. Jordan C, Drexl A (1998) Discrete lotsizing and scheduling by batch sequencing. Manag Sci 44(5):698–713
20. Kallrath J (2002) Planning and scheduling in the process industry. OR Spectrum 24(3):219–250
21. Kimms A (1996) Competitive methods for multi-level lot sizing and scheduling: Tabu search and randomized regrets. Int J Prod Res 34(8):2279–2298
22. Kimms A (1999) A genetic algorithm for multi-level, multi-machine lot sizing and scheduling. Comput Oper Res 26(8):829–848
23. Lu L (1995) A one-vendor multi-buyer integrated inventory model. Eur J Oper Res 81(2):312–323
24. Pinedo ML (2012) Scheduling: theory, algorithms, and systems. Springer, New York
25. Parija GR, Sarker BR (1999) Operations planning in a supply chain system with fixed-interval deliveries of finished goods to multiple customers. IIE Trans 31(11):1075–1082
26. Potts CN (1980) Technical analysis of a heuristic for one machine sequencing with release dates and delivery times. Oper Res 28(6):1436–1441
27. Sarker BR, Parija GR (1996) Optimal batch size and raw material ordering policy for a production system with a fixed-interval, lumpy demand delivery system. Eur J Oper Res 89(3):593–608
28. Shah N (2004) Pharmaceutical supply chains: key issues and strategies for optimisation. Comput Chem Eng 28(6):929–941
29. Sindhuchao S, Romeijn HE, Akali E, Boondiskulchok R (2005) An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity. J Glob Optim 32(1):93–118
30. Stefansson H, Sigmarsdottir S, Jensson P, Shah N (2011) Discrete and continuous time representations and mathematical models for large production scheduling problems: a case study from the pharmaceutical industry. Eur J Oper Res 215(2):383–392
31. Viswanathan S, Mathur K (1997) Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. Manag Sci 43(3):294–312
32. Viswanathan S, Piplani R (2001) Coordinating supply chain inventories through common replenishment epochs. Eur J Oper Res 129(2):277–286
33. Wang D, Grunder O, El Moudni A (2013) Single-item production-delivery scheduling problem with stage-dependent inventory costs and due date considerations. Int J Prod Res 51(3):828–846
34. Zavanella L, Zanoni S (2009) A one-vendor multi-buyer integrated production-inventory model: the consignment stock case. Int J Prod Econ 118(1):225–232
35. Zhang Y, Cao Z (2008) An asymptotic PTAS for batch scheduling with nonidentical job sizes to minimize makespan. J Comb Optim 16(2):119–126