HAL
open science

Efficient heuristics to minimize the total tardiness of chemotherapy drugs production and delivery

Alexis Robbes, Yannick Kergosien, Virginie André, Jean-Charles Billaut

To cite this version:

Alexis Robbes, Yannick Kergosien, Virginie André, Jean-Charles Billaut. Efficient heuristics to minimize the total tardiness of chemotherapy drugs production and delivery. Flexible Services and Manufacturing Journal, 2022, 10.1007/s10696-021-09431-w . hal-03345079

HAL Id: hal-03345079

https://hal.science/hal-03345079

Submitted on 23 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Efficient heuristics to minimize the total tardiness of chemotherapy drug production and delivery

Alexis Robbes · Yannick Kergosien · Virginie André · Jean-Charles Billaut

the date of receipt and acceptance should be inserted later

Abstract This study considers the production of chemotherapy drugs for cancer treatment. An important factor determining the quality of service of chemotherapy treatment is the time the patient must wait to receive his or her injection of the chemotherapy drug. Chemotherapy production and delivery are modeled as a production scheduling problem combined with a vehicle routing problem. The scheduling problem is a three-stage hybrid flow shop scheduling problem, and the routing problem is a variant of the multi-trip vehicle routing problem with due dates. The objective function is the minimization of the total time delay for chemotherapy treatment. To solve this problem, we propose several heuristic algorithms to provide quality solutions within reasonable computation times. Computational experiments are used to compare the performance of the heuristics applied to real data-based random instances.

Keywords Hybrid flow shop scheduling problem, Multi-trip vehicle routing problem, Chemotherapy production, Healthcare management, Production and distribution scheduling, Greedy randomized adaptive search procedure

Alexis Robbes, Yannick Kergosien, Jean-Charles Billaut
Université de Tours, LIFAT EA 6300, CNRS, ROOT ERL CNRS 7002, 64 avenue Jean Portalis, 37200 Tours. E-mail: {alexis.robbes, yannick.kergosien, jean-charles.billaut}@univ-tours.fr

Virginie André
CHRU de Tours, Hôpital Bretonneau, 2 boulevard Tonnellé, 37044 Tours Cedex 9
E-mail: v.andre@chu-tours.fr

1 Introduction

Healthcare is a demanding service with multiple challenges, one of which is the production and distribution of anticancer drugs. This paper discusses an integrated chemotherapy production scheduling and delivery problem, which is motivated by a practical problem faced by the Clinical Oncology Biopharmaceutical Unit (UBCO) at the regional hospital in Tours, France. The UBCO produces approximately 150 chemotherapy drugs per day and delivers them to three hospital units, with an objective of reducing patient waiting time by delivering each drug as quickly as possible. This objective is designed to improve the quality of patient care while complying with the ISO 9001 quality management certification process. To solve the production and distribution problem, this study designs and tests several algorithms and integrates the best one into decision-aid scheduling software.

The first work in collaboration with UBCO was proposed in 2010 to optimize the scheduling of the preparation of chemotherapy products (Mazier et al. 2010). The problem identified consisted of a parallel machine scheduling problem where each "machine" was a pharmacy technician. The first algorithm was implemented in decision-aid scheduling software used daily by the UBCO, and its impact was measured and analyzed by Kergosien et al. (2011). Chemotherapy drugs are produced continuously throughout the day and delivered progressively to the patients. A limited number of persons perform deliveries and delivery times are not negligible compared with production times. It is thus important to consider the production scheduling problem and the delivery problem in an integrated manner. The delivery problem was introduced by Kergosien et al. (2017), who consider a parallel machine scheduling problem with a single delivery person to deliver the drugs after production. To solve the problem, they propose an algorithm based on Benders decomposition. However, the UBCO now has several delivery persons to distribute the drugs.

To the best of our knowledge, the operations-research literature currently offers only two papers that deal with cancer-drug production and delivery. In the first of these papers, Chahed et al. (2009) address a particular issue involved in the planning of operations for home-based chemotherapy and focus on a simplified problem of chemotherapy production and delivery in the home-care context. The problem is confined to a single machine with a single delivery person who undertakes a single trip, and the objective function is the minimization of the total travel time. To solve the problem, Chahbed et al. propose a mathematical model and an exact method based on a branch-and-bound algorithm. In a second paper, Lee et al. (2014) treat the problem of producing and delivering medical treatment that requires radioactive substances with half-lives on the order of minutes. The scheduling problem is treated as a special type of bin-packing problem (machines consist of cyclotrons of varying capacities), and a fleet of heterogeneous vehicles is available for delivery. To solve the problem, Lee et al. propose a mixed integer program and a variant of a large neighborhood search algorithm, thus minimizing production costs, fixed vehicle costs, and travel costs.

Robbes et al. (2019) presented a study at the Fourth International Conference on Health Care Systems Engineering in which they consider an integrated scheduling and routing problem that represents the reality of the current UBCO production and

delivery process. The production part of the problem corresponds to a three-stage hybrid flow shop scheduling problem, where the first stage is sterilization, the second is the drug preparation process, and the final stage is the control. The routing part of the problem corresponds to a variant of the multi-trip vehicle routing problem with due dates. The objective function is the minimization of the delivery total tardiness. Robbes et al. (2019) illustrate the importance of taking into account the future-delivery problem before planning production. A multilevel heuristic is proposed to minimize the delivery total tardiness. Even if the scheduling problem and the delivery problem are considered independently, their resolution as independent problems remains a difficult problem (i.e., NP-hard).

We propose herein an extension of the treatment of Robbes et al. (2019) based on two new algorithms that significantly extend the computational experiments and outperform the algorithms used by Robbes et al. (2019).

Most of the variants of the hybrid flow shop scheduling problem are NP-hard (Abyaneh and Gholami 2015), which is also the case for the multi-trip vehicle routing problem (Cattaruzza et al. 2016). Numerous studies are available in the literature on production scheduling problems and distribution problems, but they are often solved independently. Recently, however, combined production-distribution problems have started to be studied at the operational level. For a good overview of the integration the production scheduling and vehicle routing problems, please see the review by Moons et al. (2017). In all of these studies, the scheduling problem is often based on a single machine environment (Li and Ferrell 2011; Jamili et al. 2016) or a parallel machine environment (Ullrich 2013; Belo-Filho et al. 2015), and the distribution is done with either a single vehicle (Cheref et al. 2016) or a fleet of vehicles (Li et al. 2016).

Since the work of Moons et al. (2017), new studies have emerged on the combined production-distribution problem. Table 1 lists the main studies on integrated production routing problems that share features with the problem we consider herein. For each, we give the production model, the distribution model, the main timing constraints, and the criteria to minimize. Note that the production model is also often represented by a single machine. The distribution problem that received the most attention is the Capacity vehicle routing problem, whereas few studies focus on the hybrid flow shop model for the production problem or the multi-trip aspect with several vehicles for the distribution problem.

Figure 1 shows a Venn diagram obtained from Moons et al. (2017) that displays the studies on integrated production and distribution and the studies listed in Table 1. The four sets correspond to studies that consider (i) a flow shop or job shop model, (ii) a single-machine or Parallel machines model, (iii) the multi-trip aspect (with one or several vehicles), or (iv) the vehicle routing problem (with several vehicles). The intersection of the “multi-trip” set and the “vehicle routing problem” set is a general problem, the multi-trip vehicle routing problem. The intersection of the “flow shop or job shop” set and the “Parallel machines” set is either a single-machine environment in the dotted rectangle or a hybrid flow shop model. For each possible intersection of sets the associated number of studies is displayed. The study in the thick-lined square that of Robbes et al. (2019).

Article	**Production**	**Distribution**	**Timing constraint**	**Criteria**
(Devapriya et al. 2017)	1	MT-C-VRP	Dj-Cj	D_{cost}
(Karaoğlan and Kesen 2017)	1	MT-C-TSP		D_{max}
(Kergosien et al. 2017)	P	MT-TSP	r_j, $D_j - S_j$	T_{max}
(Low et al. 2017)	1	Hete C-VRP		$D_{cost}, \sum T_j, \sum E_j$
(Fu et al. 2017)	R	Hete C-VRP	TW	P_{cost}, D_{cost}
(Miranda et al. 2018)	Lot Sizing	MT-C-TSP	TW	P_{cost}, D_{cost}
(Chekoubi et al. 2018)	Lot Sizing	CPD		P_{cost}, D_{cost}
(Gharaei and Jolai 2018)	1	C-VRP		$D_{cost}, \sum T_j$
(Lacomme et al. 2018)	1	C-VRP	Dj-Cj	D_{cost}
(Zou et al. 2018)	1	C-VRP		D_{max}
(Miranda et al. 2019)	Lot Sizing	MT-C-VRP	TW	P_{cost}, D_{cost}
(Kesen and Bektaş 2019)	P	Limited C-VRP	soft TW	$\sum T_j, \sum E_j$
(Marandi and Fatemi Ghomi 2019)	HF(P)	C-VRP	TW	D_{cost}, C_{max}
(Tamannaei and Rasti-Barzoki 2019)	1	C-VRP		$D_{cost}, \sum w_j.T_j$
(Tavares-Neto and Nagano 2019)	P	MT-C-TSP		D_{max}
(Wang et al. 2019a)	P	MT-C-VRP	TW	$D_{cost}, \sum T_j$
(Wang et al. 2019c)	HF(P)	MT-C-TSP		D_{max}
(Wang et al. 2019b)	1	C-VRP		D_{cost}
(Yağmur and Kesen 2020)	F	C-TSP		$D_{cost}, \sum T_j$
(Ganji et al. 2020)	1	Hete C-VRP		$D_{cost}, \sum T_j$, Preference
(Liu and Liu 2020)	1	C-VRP		$\sum w_j.D_j$
(Liu et al. 2020)	1	C-VRP		$\sum D_j$
(Mohammadi et al. 2020)	J	C-VRP		$P_{cost}, D_{cost}, \sum T_j, \sum E_j$
(Robbes et al. 2019) and this paper	HF(P)	MT-VRP	r_j	$\sum T_j$

Table 1 Overview of main recent studies on combined production-distribution problem since (Moons et al. 2017). 1: single machine; P: parallel machine; R: unrelated parallel machine; J: job shop; F: flow shop; HF(P): hybrid flow shop and parallel machine; Hete: heterogeneous; MT: multi-trip; C: capacity; TSP: traveling salesman problem; VRP: vehicle routing problem; TW: time windows; D_j: delivery time; r_j: release date; S_j: production starting time; C_j: production completion time; E_j: earliness; T_j: tardiness; w_j: weight; P_{cost}: production costs (setup costs, processing costs, inventory costs); D_{cost}: distribution Cost (travel costs with or without vehicle fixed costs)

We now briefly review some papers that deal with a similar problem. Kesen and Bektaş (2019) tackle an integrated Parallel machine scheduling problem and a vehicle routing problem with Time Windows by proposing two soft time window variants, depending on whether idle waits are allowed. Both variants are modeled by a mixed integer model, and the sum of total earliness and total tardiness are minimized. A limited homogeneous fleet of vehicles of limited capacity ensures the delivery. Wang et al. (2019a) discuss an integrated production and multi-trip vehicle routing problem with Time Windows and uncertain travel times. The production environment consists of Parallel machines with machine-dependent ready times, and deliveries are made by a fleet of identical vehicles. The objective is to minimize two types of costs: the travel cost and penalty cost due to tardiness. Wang et al. (2019a) propose a robustness approach based on the elastic p-robustness to deal with travel time variations and a memetic algorithm to solve the problem. Computational experiments to evaluate the proposed approach were conducted on instances with up to 50 jobs. In other work, Yağmur and Kesen (2020) studied a permutation flow shop and vehicle routing prob-

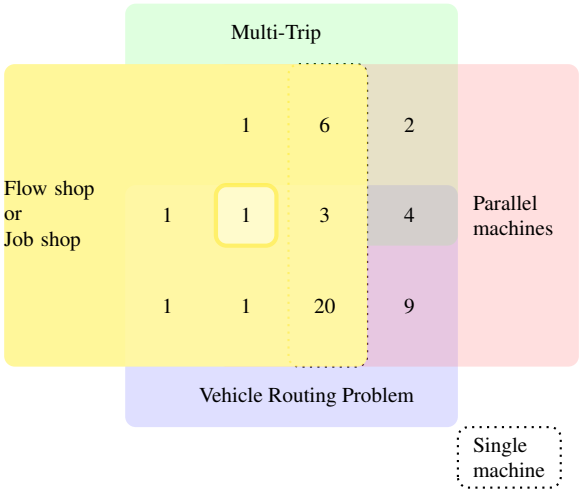

Fig. 1 Venn diagram of state of the art for integrated production and distribution state

lem. In a flow shop environment, a set of orders are produced on machines and are delivered using a single vehicle. However, orders are produced in batches, and all orders in a batch are delivered on the same trip. The authors develop a memetic algorithm with different decoding strategies to solve the problem, thus minimizing the sum of total trip time and total tardiness. Finally, the study of Mohammadi et al. (2020) is inspired by a case study of a furniture manufacturing company. The paper discusses a bi-objective mixed integer model of an integrated flexible job shop scheduling problem and vehicle routing problem with Time Windows with a heterogeneous fleet. The two objectives are the total operational costs based on the production cost and delivery cost, and the weighted sum of delivery earliness and tardiness. The authors proposed an ε-constraint method and a hybrid particle swarm optimization heuristic.

Figure 2 shows the distribution of the studies listed in Table 1 and that of Moons et al. (2017) as a function of the most common objective functions to minimize: the makespan, the maximum delivery time, the total delivery time, the total tardiness, and the distribution costs (i.e., travel costs with or without vehicle fixed costs). The objective function most often considered (around 60% of the studies) is the minimization of distribution costs. The objective function considered herein (minimization of total tardiness) has been addressed in 12 studies, although it is not overly common in the literature on the hybrid flow shop scheduling problem (Tosun et al. 2020; Ruiz and Vázquez-Rodríguez 2010) and on the multi-trip vehicle routing problem, with only three papers considering this objective function (Liberatore et al. 2011; Shelbourne et al. 2017; Karoonsoontawong et al. 2020).

The present paper discusses an integrated chemotherapy drug production and delivery problem (Robbes et al. 2019) that is based on a real-life case. The collaboration with Tours hospital goes back 15 years now, with a first article published in 2010 (Mazier et al. 2010). A software has been developed and is in use every day since then (Kergosien et al. 2011). All steps of modeling process, including the assumptions on

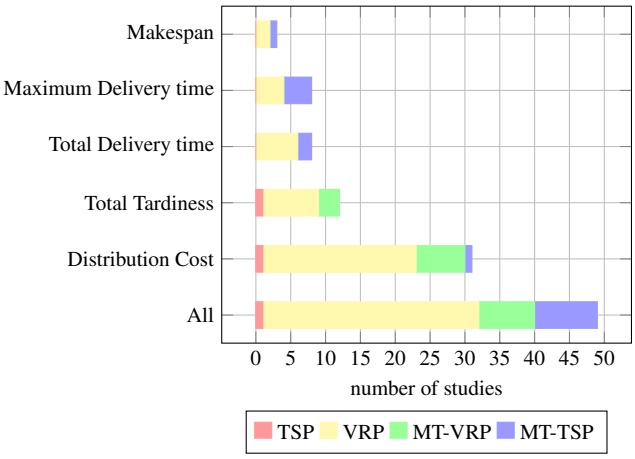

Fig. 2 Objective functions in integrated production and distribution problem

the data made, were carried out in collaboration with the hospital unit (UBCO). Note that other services such as UBCO have a similar production and distribution environment. The problem to solve is original and complex because it integrates several Operational Research problems which are scheduling and routing. These problems are generally solved separately, and are already difficult separately. This type of integrated problems are relatively recent in the literature. Moreover few studies consider the minimization of total tardiness with the multiple trips features. The contribution of this paper is the proposition of two new solving methods: a bilevel gradient descent (BGD) and a greedy randomized adaptive search procedure (GRASP) based on the multilevel heuristic from Robbes et al. (2019). These new methods provide good solutions within a reasonable time and are destined to be integrated into a decision-making tool and to serve as an online algorithm every time a new event occurs (e.g., the arrival of a new production request). One originality in the method proposed to solve the problem is that it integrates clustering methods and more traditional algorithms for scheduling and routing problems, within an iterative method (scheduling-routing-scheduling-routing...). The computational experiments have been significantly extended to compare all the resolution methods and to show that the new methods outperform those presented by Robbes et al. (2019).

The remainder of this paper is organized as follows. Section 2 formally defines the problem and presents the notation. All model assumptions have been validated by the UBCO. The resolution methods are presented in Section 3, and then Section 4 describes the computational experiments and results. Finally, Section 5 presents the conclusion and future research directions.

2 Problem statement and notations

2.1 Production and delivery processes

Chemotherapy injections are manufactured in a regularly cleaned sterile environment with filtered air and by operators with specialized clothing. The preparation process is safeguarded by applying several methods that enhance the quality and security of the chemotherapy treatment (Savelli et al. 2018).

Producing an injectable chemotherapy treatment is done in several stages. First, the patient receives a medical consultation a few days before the scheduled date of his or her treatment. After this consultation, the doctor prescribes the treatment to be administered (type of molecule, dosage, etc.). A production order for the prescribed medicines is sent to the UBCO service, and another consultation is scheduled with the doctor a few moments before the onset of treatment. During this second consultation, the doctor checks the patient's health, validates the previous prescription, and authorizes the chemotherapy injection. The preparation of the drug by the UBCO service can start only once this validation is complete. The goal of this procedure is to avoid loss of medication when the patient's health precludes receiving the medication. Figure 3 illustrates this process.

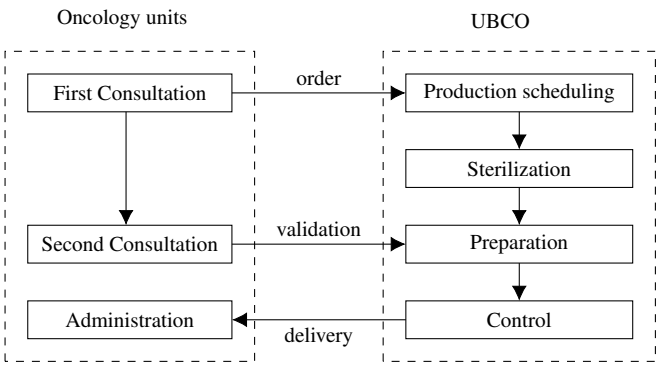

Fig. 3 Chemotherapy treatment process

The chemotherapy production is a three-step process:

1. `Sterilization` by batch of the baskets containing all the required material to make the preparations (one basket per chemotherapy drug);
2. `Preparation` of the drug by a pharmacy technician;
3. `Control` of a sample of the drug.

The `Sterilization` and `Preparation` steps are done by using a machine called an "isolator." Each isolator is a sealed unit that protects the operator from the cytotoxic drugs, which are classified as hazardous to the health and potentially carcinogenic. The sealed units also protect the drugs from microbiological contamination during production. Several workstations are connected to the sterilizer and are

supplied with purified air via a configuration specific to the isolator. The production of a drug can start by sterilizing its basket before the treatment is validated by the doctor during the second consultation. If the treatment is not validated, the sterilized products are returned to the storage area because they have not been used and the sterilization step does not degrade them. Otherwise, the production process continues. The pharmacy technicians (also called operators) execute the `Preparation` step by hand. The duration of this step depends on the type of medication being prepared. Once the chemotherapy preparation is complete, the `Control` step is executed to enhance the security and quality of the drugs. Each chemotherapy drug is intended for a given patient, and all patients are situated in different oncology units of the hospital.

After production, the drugs are distributed throughout the day by a team composed of a constant number of delivery persons. The size of this team cannot be changed for political reasons internal to the hospital. The main objective of the UBCO is to produce and deliver the injectable drugs on time and to continually improve the quality management system, as required by the ISO 9001 standards. However, this goal is rarely attainable because of the high workload and because the drugs cannot be prepared in advance. Thus, the UBCO decided to use total tardiness as one of its main performance indicators (other indicators related to traceability and safety are also very important). Note that production costs are constant and the delivery costs (i.e. travel costs) are relatively unimportant compared with the importance of the timely delivery of medications.

2.2 Problem definition and notations

We model the chemotherapy production and delivery problem as an integrated scheduling and routing problem composed of a three-stage hybrid flow shop scheduling problem for the scheduling part, and a variant of the multi-trip vehicle routing problem with due dates for the routing part. Each chemotherapy drug intended for a single patient is called a job. In the hybrid flow shop scheduling problem, a set of jobs has to be processed in a series of production stages, where each stage may have multiple machines operating in parallel. Once jobs are processed, they must be delivered, which implies solving a multi-trip vehicle routing problem. This problem consists of finding a set of routes for each vehicle involved in job delivery. Each route is associated with a subset of jobs to be delivered and starts and ends at the production site.

We now describe the input parameters. The production site is composed of a set $\mathscr{I}$ of identical parallel isolators. Each isolator $I_i \in \mathscr{I}$ is characterized by the sterilizer capacity Q_i (i.e., the maximum number of jobs that can be sterilized at a time), a constant sterilization processing time p^S (which does not depend on the job or the number of jobs in the sterilized batch), and a number m_i of operators that are connected to the isolator I_i and can work at the same time (i.e., the number of workstations). Figure 4 shows one isolator used by the UBCO: it is composed of a sterilizer that can sterilize up to twelve baskets simultaneously and two face-to-face workstations where the operators work using a glovebox system.

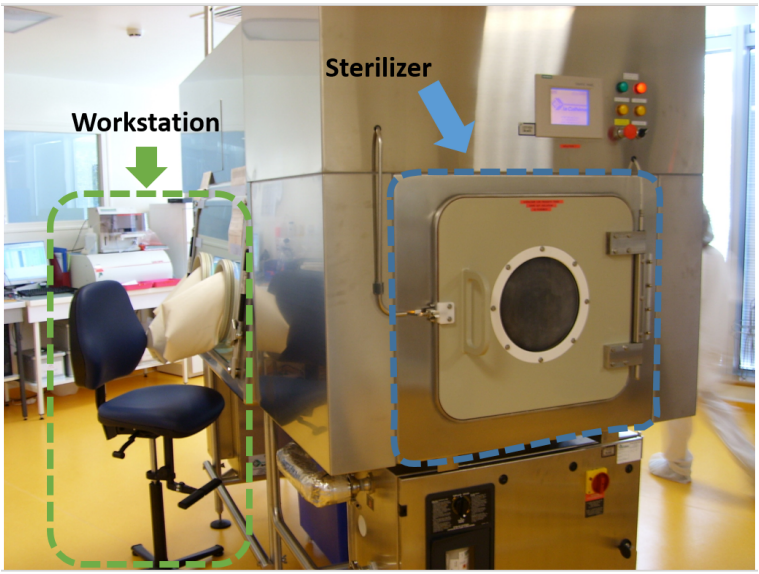

Fig. 4 Isolator picture: operators work outside of the isolator using a glove box system.

The set of chemotherapy drugs to be produced and delivered is represented by the set $\mathscr{J}$ of jobs. All jobs have the same production routing in the manufacturing site. Each job is assigned to one isolator (first stage of the hybrid flow shop) and to one of the workstations associated with the isolator (second stage). In stage three, each job is processed on a control machine. The control step is performed by a single automated analyzer, and jobs cannot be preempted.

Each job $J_j \in \mathscr{J}$ is characterized by a release date r_j corresponding to the validation date before which the preparation step cannot begin, a processing time p_j^O corresponding to the time required by the `Preparation` step, an assigned oncology unit u_j where it has to be delivered, and a delivery due date d_j. The control processing time, denoted p^A, is the same for every job because this operation is the same for every chemotherapy drug.

The jobs are then grouped in several batches for delivery. Each delivery batch represents a set of jobs that are delivered on the same trip. Each trip is assigned to a delivery person who can only leave the production site once all jobs in the delivery batch are controlled. We denote by $tt_{j,j'}$ the travel time between units u_j and $u_{j'}$. Each delivery person can make several trips and has unlimited capacity. A trip corresponds to (1) leaving the production site, (2) delivering a set of jobs in several units, and (3) returning to the production site.

The objective function is the minimization of the total tardiness of delivery, which is given by

$$\text{MIN} \sum_{J_j \in \mathscr{J}} T_j,$$

where T_j is the delivery tardiness of job J_j and is defined by

$$T_j = \max(0, D_j - d_j), \tag{1}$$

where D_j is the delivery date of J_j.

We use the following notation:

- C_j^O is the completion time of job J_j in the preparation step;
- C_j^A is the completion time of job J_j in the control step.

The problem consists in finding the sequences and composition of the sterilization batches on each isolator, the production sequences and starting times of the jobs on each operator and on the analyzer, the composition of each delivery batch, and the sequences and composition of routes for each delivery person. After returning to the production site, the delivery person starts her next route as soon as the last job of her delivery batch is completed. A mathematical formulation of this process is given in Appendix A.

2.3 Illustration

Figure 5 shows a Gantt chart representing a partial solution to a problem instance with $|\mathscr{I}| = 2$ isolators, $m_i = 2$ operators, a sterilizer capacity $Q_i = 4$ for each isolator, and $|V| = 2$ delivery persons.

As an example, let us follow the process of job J_{20} from the `Sterilization` step to its delivery. First, job J_{20} is sterilized in the first batch of isolator I_1. After completion of the `Sterilization` step and of Job J_1, and after its release date r_{20}, job J_{20} is prepared by operator $O_1 = 1$. It is then scheduled in position 5 in the analyzer after job J_5. Its control starts after its preparation-completion time C_{20}^O. Next, job J_{20} is packed in a delivery batch with jobs J_9, J_{11}, and J_{12}, and the batch is assigned to delivery person 2 and is ready to leave the production site at time $\max(C_9^A, C_{11}^A, C_{12}^A, C_{20}^A)$.

The delivery trip containing job J_{20} is the first delivery of delivery person 2, who leaves the UBCO service after the completion time of the last job of the batch (job J_{12} in this case) to deliver job J_9 to its assigned unit u_9, following which she leaves this unit to deliver jobs J_{11} and J_{12} to their assigned unit $(u_{11} = u_{12})$. Job J_{20} is delivered last at its delivery unit u_{20}. Finally, the delivery person returns to the UBCO site and is available for the next trip.

3 Resolution methods

This section presents three deterministic heuristic algorithms and a GRASP. The choice to develop such heuristics is motivated by the goal to have a fast and efficient resolution method to be integrated into a real-time decision-making tool.

Each deterministic heuristic contains three decision levels: the assignment of jobs to sterilization batches, the production sequencing and scheduling of jobs, and the delivery planning.

The first two heuristics are based on those presented by Robbes et al. (2019), and two new heuristic algorithms are proposed. The first heuristic, called the reference algorithm (RA), is a simple greedy method that consists of taking decisions level by

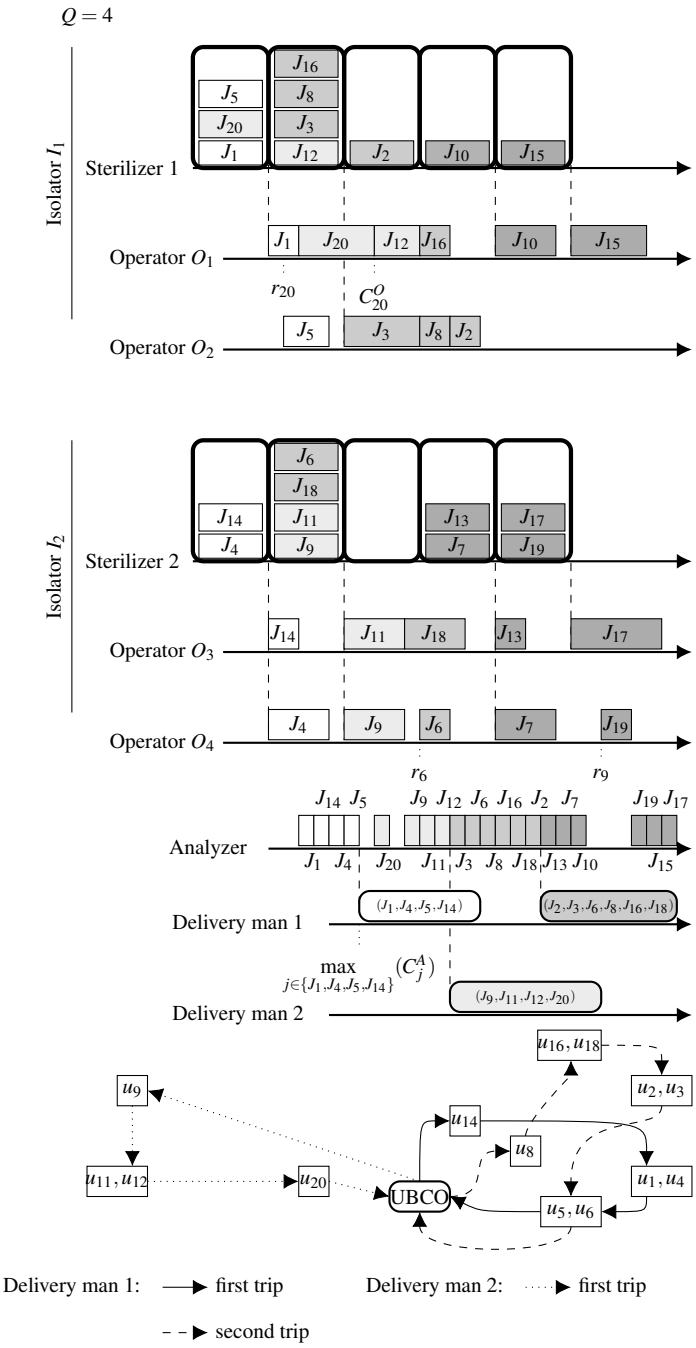

Fig. 5 Illustration of production-delivery problem.

level (from the assignment of jobs to the sterilization of batches up to the delivery planning). The idea of the second heuristic, called the multilevel (ML) heuristic, is to improve the previous heuristic by taking into account the data routing of the first decision level. The first two heuristics RA and ML have as input a fixed number of delivery batches. The last two resolution methods are an extension of the ML heuristic and use a gradient descent procedure to automatically determine the optimal number of delivery batches. The third heuristic is called bilevel gradient descent (BGD). The fourth heuristic is the greedy randomized adaptive search procedure (GRASP), which has the same structure as the bilevel gradient descent plus a diversification phase.

Figure 6 shows the three deterministic heuristics in the form of a flowchart that presents the main steps of each algorithm according to each level of decision. We denote by n_f the number of final delivery batches used to compute the *Final clustering* step; n_f also represents the total number of trips assigned to the delivery team. n_t is the number of target or desired delivery batches used to compute the *Production scheduling*. The algorithms are now presented in detail.

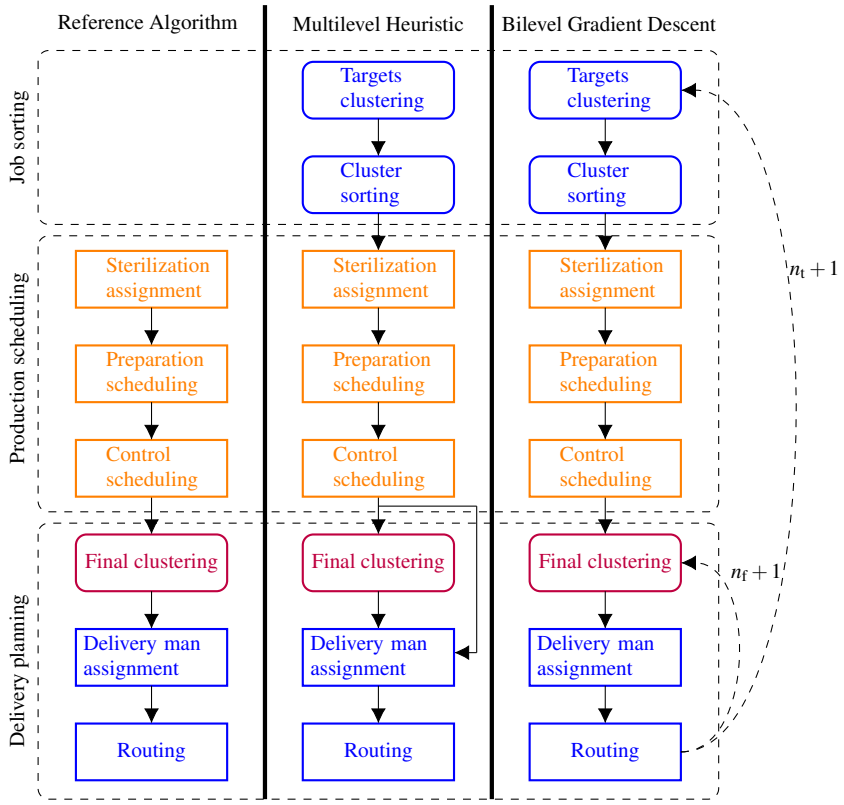

Fig. 6 Flowchart of reference algorithm, multilevel heuristic, and bilevel gradient descent.

3.1 Reference Algorithm

The RA is inspired from the production and delivery process used at the UBCO site. The decisions are taken naturally based on the following steps: First, the jobs are assigned to the sterilization batches of an isolator. Next, they are scheduled to one of the associated operators and on the common control machine. Finally, they are assigned to a trip executed by a delivery person.

No idle time elapses between two consecutive sterilization batches because the sterilization of jobs is anticipated before their release dates. Therefore, we assume that a sterilization batch runs every p^S times on each isolator $I_i \in \mathscr{I}$. The `Sterilization` completion time of batch k on isolator I_i (denoted $B_{i,k}$) is kp^S. The first decision is to assign each job to a sterilization batch (*Sterilization assignment*). The following procedure is used to find the composition of the sterilized batches. First, the jobs are sorted in increasing order of release dates. Next, each job J_j is successively assigned to a sterilization batch that is not full and according to the following rules:

- If none of the sterilization batches completing before r_j are full, job J_j is assigned to a sterilization batch that will complete its sterilization as late as possible but before the release date r_j of J_j.
- Otherwise, job J_j is assigned to the earliest available sterilization batch completing after the release date.
- Given that several isolators are available, if there is more than one candidate sterilization batch, the selected batch is the batch with the minimum total `Preparation` processing time for jobs in the batch (i.e., the batch $B_{i,k}$ that minimizes $\sum_{J_j \in B_{i,k}} p_j^O$).

The resulting assignment to sterilization batches is the input of the next step, called *Preparation scheduling*. Sterilization batch by sterilization batch, jobs are sorted by increasing release date and are assigned to the first available machine (i.e., operator) of the corresponding isolator.

The last scheduling decision is the *Control scheduling*. The jobs are prepared and then assigned to the control machine in first-in-first-out order (i.e., by respecting the `Preparation` completion time order).

Finally, the last decision level concerns the delivery planning. Each job is first assigned to a delivery batch by a *Final clustering*, and each delivery batch is assigned to a delivery person. *Final clustering* aims to limit the time that the job has to wait before delivery and to reduce the trip duration. It is therefore based on the job due dates, on the job `Control` completion time, and on the delivery locations.

The clustering algorithm is an agglomerative hierarchical clustering method (Murtagh and Contreras 2012) and is described by the pseudocode in Alg. 1. This method requires that a distance function be defined between two jobs and between two clusters. We define the distance $dist_{\text{clusters}}(E, E', dist_{\text{f}})$ between clusters E and E' [also called a "complete link" by Murtagh and Contreras (2012)] as the maximum distance between jobs J_j and $J_{j'}$, which is denoted $dist_{\text{f}}(j, j')$:

$$dist_{\text{clusters}}(E, E', dist_{\text{f}}) = \max_{J_j \in E,\, J_{j'} \in E'} \big(dist_{\text{f}}(j, j')\big). \tag{2}$$

The distance between jobs J_j and $J_{j'}$ is the three-dimensional Euclidean distance with time in minutes, where the three dimensions are the travel time $tt_{j,j'}$ between oncology unit locations, the due dates d_j and $d_{j'}$, and the `Control` completion times C_j^A and $C_{j'}^A$. The distance between jobs j and j' is thus

$$dist_{\mathrm{f}}(j,j') = \sqrt{\left(d_j - d_{j'}\right)^2 + \left(C_j^A - C_{j'}^A\right)^2 + tt_{j,j'}^2}. \tag{3}$$

The desired number n_{f} of clusters should be given as an input to the clustering algorithm. This important parameter is related to the maximum acceptable distance between jobs from the same delivery cluster. The batching problem is solved by calling `AH_Clustering(`$\mathscr{J}, n_{\mathrm{f}}, dist_{\mathrm{f}}$`)`.

```
Parameter: set of jobs 𝒥, number of delivery batches n_f and distance function dist_f
Let ℰ be a partition of 𝒥 composed only of singletons
while |ℰ| > n_f do
    best_dist ← +∞
    for any E ∈ ℰ do
        for any E' ∈ ℰ \ {E} do
            if dist_clusters(E, E', dist_f) < best_dist then
                best_dist ← dist_clusters(E, E', dist_f)
                best_E ← E
                best_E' ← E'
            end if
        end for
    end for
    ℰ ← ℰ \ {best_E, best_E'}
    best_E ← best_E ∪ best_E'
    ℰ ← ℰ ∪ {best_E}
end while
return ℰ
```

Algorithm 1: `AH_Clustering(`$\mathscr{J}, n_{\mathrm{f}}, dist_{\mathrm{f}}$`)`

Figure 7 illustrates the agglomerative hierarchical clustering method using a dendrogram graphic. The length of the vertical lines represents the distance between two clusters. The dendrogram construction starts from the bottom, where each cluster is composed of only one job and stops upon reaching n_{f}. The dashed line represents the cutoff of the dendrogram and determines the resulting clusters of the algorithm: $\{J_1, J_4, J_5, J_{14}\}$, $\{J_9, J_{11}, J_{12}, J_{20}\}$, and $\{J_2, J_3, J_6, J_8, J_{16}, J_{18}\}$.

The last steps are the assignment of delivery batches to a delivery person (*Delivery person assignment*) and the determination of the *Routing* of each batch. The resulting delivery batches are assigned according to the increasing order of their completion times (i.e., the maximum completion time of all jobs in the batch) to the first available delivery man. The routes of each delivery person minimize the total tardiness of the jobs (the objective function of the problem) and the length of the routes are also minimized. This second objective also aims to reduce the total tardiness of jobs in the forthcoming routes of the same delivery person because the sooner a delivery person returns to the production site, the sooner he can start his next route. Thus, to

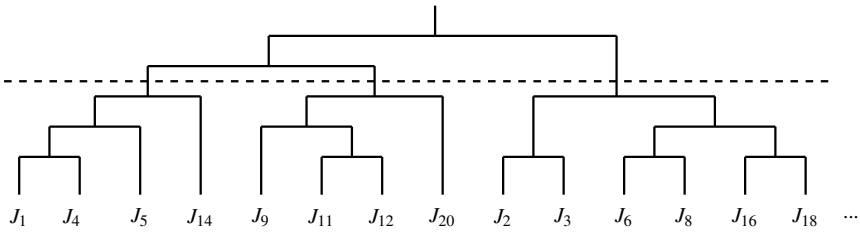

Fig. 7 Dendrogram diagram of execution of Alg. 2 on the example of Fig. 5.

compute the routes in a short computation time, we use the nearest-neighbor heuristic (Rosenkrantz et al. 1977) to build the routes of each delivery person.

3.2 Multilevel heuristic algorithm

The ML heuristic uses a clustering algorithm and dispatching rules before scheduling the jobs on isolators. The main steps are detailed by Robbes et al. (2019). The clustering method is used to define target delivery batches *a priori* (i.e., before calculating the production schedule). This procedure aims to schedule production by taking into account not only the release dates, as in the previous method, but also the due dates and locations to which jobs are to be delivered.

The clustering method used to create the target delivery batches is the `AH_Clustering` algorithm presented in Alg. 1. We introduce a *revised release date* $\widetilde{r}_j = r_j + p_j^O$ that corresponds to the earliest possible preparation-completion time. The problem is solved by calling AH_Clustering($\mathscr{J}, n_\mathrm{t}, dist_\mathrm{t}$) with a new distance function $dist_\mathrm{t}(j, j')$ given by

$$dist_\mathrm{t}(j, j') = \sqrt{\left(d_j - d_{j'}\right)^2 + \left(\widetilde{r}_j - \widetilde{r}_{j'}\right)^2 + tt_{j,j'}^2} \tag{4}$$

The purpose of creating the target delivery batches is to provide a more suitable *Job sorting* by applying *Cluster sorting*. Each resulting target delivery batch is associated with a due date that is the minimum due date of the jobs in the batch. The target delivery batches are sorted according to their earliest due date. As before, the number n_t of the desired clusters or target delivery batches should be given as an input of the clustering algorithm.

The *Sterilization assignment* of the jobs uses `Assign_ster_batches`(σ), where σ is the list of jobs sorted first according to the target delivery batch order. Inside a batch, the jobs are sorted in increasing order according to their earliest revised release date. Next, *Preparation scheduling* and *Control scheduling* are done in the same way as in the RA, two batching solutions (i.e., assignment of jobs to delivery batches) are computed, and the best solution is selected. The first batching solution is based on the target delivery batches and consists of the same batches as calculated in the first step. The second batching solution aims to determine new batches using the *Final clustering* method and the $dist_\mathrm{f}$ function as the distance between two batches. The value n_f for the *Final clustering* is set equal to n_t, which implies that, in most

cases, the final batches should be the same as the target batches. Each batching solution is evaluated by the same *Delivery person assignment* and *Routing* methods as in the RA.

3.3 Bilevel gradient descent

Bilevel gradient descent (BGD) is an improved, self-designed version of the ML heuristic. Without testing all the values, BGD searches for the couple $(n_\mathrm{t}, n_\mathrm{f})$ that leads to the best solution. Robbes et al. (2019) presented experiments that demonstrate that the best solution evolves as a function of the number $n_\mathrm{t} = n_\mathrm{f}$ of delivery batches, which is a curve with a single minimum turning point. A method such as a gradient descent (Snyman 2005; Ruder 2016) is, therefore, particularly well suited to find a good number of delivery batches. Further experiments demonstrated that the exact value of this tipping point depends on the instance and that using two different values n_t and n_f may improve the results of the heuristic algorithm. To address this setting optimization problem, we propose a "bilevel gradient descent heuristic" that combines two gradient descent algorithms. This heuristic uses a main gradient descent `BGD` to optimize the number n_t of target delivery batches. In addition, for each value tested of n_t, a secondary gradient descent `sec_GD` is called to optimize the number n_f of final delivery batches. The BGD uses two methods: one for *Production scheduling* and one for *Delivery planning*.

Before presenting the algorithms, we introduce the following definitions:

- Let $PS : \mathscr{E} \mapsto \mathscr{S}$ be the *Production scheduling* method that returns a production schedule $\mathscr{S}$ based on the set $\mathscr{E}$ of target batches ($|\mathscr{E}| = n_\mathrm{t}$). The production schedule defines the sequences and composition of the sterilization batches on each isolator, the production sequences and starting times of the jobs on each operator and on the analyzer. The *PS* method consists of successfully applying the *Sterilization assignment*, *Preparation scheduling*, and *Control scheduling* methods used in the RA and the ML heuristic.
- Let $DP : (\mathscr{S}, n_\mathrm{f}) \mapsto \mathscr{S}'$ be the method that builds the *Delivery planning* $\mathscr{S}'$ associated with the production schedule $\mathscr{S}$ with a number of delivery batches n_f. A delivery planning defines the composition of each delivery batch, and the sequences and composition of routes of each delivery person. The *DP* method consists of successfully applying the *Final clustering*, *Deliver assignment*, and *Routing* methods.
- Let $TT : \mathscr{S}' \mapsto z$ be a function that computes the total tardiness associated with delivery planning $\mathscr{S}'$.

The pseudo codes for `BGD` and `sec_GD` are presented in Algs. 2 and 3, respectively.

Parameter: set of jobs $\mathscr{J}$, distance $dist_{\mathrm{t}}$ and learning rates α_{main} and α_{sec}
$n_t \leftarrow 1$
$\mathscr{E} \leftarrow \texttt{AH_clustering}(\mathscr{J}, n_t, dist_{\mathrm{t}})$
$\mathscr{S}_{n_t} \leftarrow PS(\mathscr{E})$
$\nabla_{main} \leftarrow \texttt{sec_GD}(\mathscr{S}_{n_t}, \alpha_{\mathrm{sec}})$
best_value $\leftarrow \nabla_{main}$
while $\nabla_{main} \leq 0$ **do**
  $\mathscr{E} \leftarrow \texttt{AH_clustering}(\mathscr{J}, n_t + 1, dist_{\mathrm{t}})$
  $\mathscr{S}_{n_t+1} \leftarrow PS(\mathscr{E})$
  $\nabla_{main} \leftarrow \alpha_{\mathrm{main}} \cdot \nabla_{main} + \texttt{sec_GD}(\mathscr{S}_{n_t+1}, \alpha_{\mathrm{sec}}) - \texttt{sec_GD}(\mathscr{S}_{n_t}, \alpha_{\mathrm{sec}})$
  best_value $\leftarrow$ min(best_value, $\texttt{sec_GD}(\mathscr{S}_{n_t+1}, \alpha_{\mathrm{sec}})$)
  $n_t \leftarrow n_{\mathrm{t}} + 1$
  $\mathscr{S}_{n_t} \leftarrow \mathscr{S}_{n_t+1}$
end while
return best_value

Algorithm 2: $\texttt{BGD}(\mathscr{J}, dist_{\mathtt{t}}, \alpha_{\mathtt{main}}, \alpha_{\mathtt{sec}})$

Parameter: production schedule $\mathscr{S}$ and learning rate α_{sec}
$n_f \leftarrow |V|$
$\nabla_{sec} \leftarrow -TT(DP(\mathscr{S}, n_f))$
best_value $\leftarrow TT(DP(\mathscr{S}, n_f))$
while $\nabla_{sec} \leq 0$ **do**
  $\nabla_{sec} \leftarrow \alpha_{\mathrm{sec}} \cdot \nabla_{sec} + TT(DP(\mathscr{S}, n_f + 1)) - TT(DP(\mathscr{S}, n_f))$
  best_value $\leftarrow$ min(best_value, $TT(DP(\mathscr{S}, n_f + 1))$)
  $n_f \leftarrow n_{\mathrm{f}} + 1$
end while
return best_value

Algorithm 3: $\texttt{sec_GD}(\mathscr{S}, \alpha_{\mathtt{sec}})$

The BGD algorithm takes four parameters as input: the set $\mathscr{J}$ of jobs, the distance $dist_{\mathrm{t}}$ used by the clustering method, and two learning rates (one for each gradient descent). The BGD heuristic is based on a classic gradient descent that searches for the best target number n_{t} for the delivery batch that leads to the best solution to the problem. The method starts at $n_{\mathrm{t}} = 1$ and moves from n_{t} to $n_{\mathrm{t}} + 1$ as many times as needed, with a solution computed for each value n_{t}. The stopping criterion depends on the current solution, the solution computed at the previous iteration, and the learning rate α_{main}. For each tested value of n_{t}, a solution is computed in three steps: First, a set $\mathscr{E}$ of n_{t} clusters is identified by using the AH_Clustering algorithm. Next, the *PS* method is called to find a production schedule $\mathscr{S}_{n_{\mathrm{t}}}$. Finally, the solution is completed by computing the delivery planning using the secondary gradient descent sec_GD that searches for the best final number n_{f} of delivery batches and returns the value of the objective function for the best solution found. The sec_GD algorithm is also based on a classic gradient descent that starts at $n_{\mathrm{f}} = |V|$ (the number of delivery persons) and continues iterating n_{f} until a stopping criterion is reached. From a production schedule $\mathscr{S}$ and for each tested value n_{f}, the *DP* method completes the solution by determining the delivery plan. Finally, TT returns the value of the objective function.

Note that the AH_Clustering algorithm that builds the dendrogram is applied only one time at the beginning of the execution of the BGD heuristic. The exploration strategy is thus similar to a "top-down" approach, which consists of starting from the top of the dendrogram (i.e., a single cluster) and successively reversing each agglomerative step.

The learning rates α_{main} and α_{sec} fall in the range $[0,1]$. The purpose of a positive learning rate is to overcome hypothetical local minima, and the purpose of a learning rate less than unity is to avoid steps that decrease excessively after passing the turning point.

3.4 Greedy randomized adaptive search procedure

The UBCO strategy is given by an online algorithm that updates the solution approximately every minute. As the computational experiments show, the proposed BGD runs fast for a full-day schedule, which leaves room for improvement. We thus develop a GRASP (Resende and Ribeiro 2014) based on the BGD to improve the quality of the solution by using the available computation time to the fullest extent. The GRASP is based on the following principles: first, a pool of greedy randomized solutions is generated, following which the solutions are improved with a local search, and then the best solution found is returned. The proposed GRASP consists of injecting some randomness into the *Target clustering* step and using the BGD as an improvement phase. To inject randomness into the construction of batches, we use the new distance function

$$dist_{\mathrm{random}}(\theta, j, j') = [1 + \mathrm{random}(-\theta, \theta)]dist_{\mathrm{t}}, \tag{5}$$

where $\theta \in [0,1]$ represents the percent error allowed in the distance computation. The pseudo code for GRASP is presented in Alg. 4. The initial iteration of the GRASP is

the original BGD (without randomization), the number n_{iter} of iterations represents the number of randomized solutions.

```
Parameter: set of jobs 𝒥, error percentage θ and learning rates α_main
and α_sec and the number of iterations n_iter
best_value ← BGD ( 𝒥, dist_t, α_main, α_sec )
k ← 1
while k ≤ n_iter do
    best_value ← min(best_value, BGD ( 𝒥, dist_random, α_main, α_sec ) )
    k ← k + 1
end while
return best_value
```

Algorithm 4: GRASP ($\mathscr{J}, \theta, \alpha_{main}, \alpha_{sec}, n_{iter}$)

4 Computational experiments

To test the effectiveness of the methods in a more general context (i.e., in other production units of chemotherapy drugs), we evaluate the proposed heuristics by applying them to two types of instances: one type based on real data from UBCO (called "UBCO instances") and one type of random instances (called "random instances"). The chemotherapy drug production process is essentially the same regardless of the hospital. However, the number of workstations per isolator, the sterilizer capacity, and the processing times may change from one unit to another. Moreover, a production unit can deliver to more than three hospitals, especially in the case of home chemotherapy. Table 2 presents the parameter settings used to generate the data sets. The first rows of the table correspond to the common characteristics (production capacities). For each instance type, we generate several sets of instances characterized by a number of chemotherapy drugs and a number of delivery persons. Each set receives 20 instances, so a total of 480 instances are generated, 240 instances for each type. Each graphic and table of this section is based on the same 20 instances of UBCO type or random type.

All experiments were done on a Dell laptop equipped with an Intel(R) Core™ i5-7440HQ CPU (2.80 GHz), 8 GB main memory, and Windows 10. All heuristics were coded in C++ and executed on a single thread.

4.1 Settings for reference algorithm and multilevel heuristic algorithms

The purpose of this experiment is to determine the "ideal" value of the number n_{f} of final delivery batches used for the *Final clustering*, which itself depends on the number $|V|$ of delivery persons and the number $|\mathscr{J}|$ of chemotherapy drugs. To obtain the best value of the parameter n_{f}, we use the RA and ML heuristic to solve all instances of each type and for each possible value of $n_{\text{f}} \in \{|V|, \ldots, |\mathscr{J}|\}$ and $n_{\text{t}} = n_{\text{f}}$. For each instance, the minimum number of delivery batches that minimizes the total tardiness

	UBCO instances	**Random instances**
Number of chemotherapy drugs	$\lvert\mathscr{J}\rvert \in \{120, 140, 160, 180\}$	
Number of delivery persons	$\lvert V\rvert \in \{3, 4, 5\}$	
Control processing time	$p^A = 3$ minutes	
Sterilization processing time	$p^S = 15$ minutes	
Due dates horizon	$d_j \in [60, 600]$ minutes	
Number of isolators	$\lvert I\rvert = 3$	
Number of operators per isolator	$m_i = 2$ operators for any isolator	
Processing times	$p_j^O \in \{5, 10, 15\}$ minutes	$p_j^O \in [5, 20]$ minutes
Release dates horizon	$r_j \in \{d_j - 90, d_j - 60\}$ minutes	$r_j \in [0, d_j - 60]$ minutes
Number of hospitals	3	$\lvert\mathscr{J}\rvert$
Travel times	The UBCO is located in a hospital, about 15 minutes from a second hospital and about 35 minutes from the third hospital. In each hospital, between 0 and 5 additional minutes are needed to reach the delivery unit.	The delivery points are distributed in a square around the production site that is positioned in the middle. Each point is within 40 minutes of the production unit.

Table 2 Instances characteristics

is retained. Figure 8 shows all the results in the form of boxplots in four different graphics for the varying instance types and algorithms. Each boxplot corresponds to the best value for the number n_f of delivery batches for each set of 20 instances.

The experimental results shown in Fig. 8 highlight how the number of delivery persons affects the optimal number of delivery batches. However, the size of the instance (i.e., the number of chemotherapy drugs) does not significantly affect the optimal number of delivery batches. The median best value for n_f is also clearly related to the instance type (UBCO or random). The median best number of delivery batches is three times greater for the UBCO instances than for the random instances. This result is explained primarily by the fact that, for UBCO instances, the production unit is located at a single hospital. Thus, a third of the total drugs can be delivered to locations within five minutes of the production site, which favors several short delivery trips.

For the rest of the computational experiments, the number of the final delivery batches for RA and ML heuristic are fixed as follows:

- $n_\mathrm{f} = 11(|V| - 1)$ for UBCO instances;
- $n_\mathrm{f} = 3(|V| - 1)$ for random instances;
- $n_\mathrm{t} = n_\mathrm{f}$ for the ML heuristic.

These values are represented by dotted lines in Fig. 8.

4.2 Evaluation of heuristic algorithms

To compare all the proposed heuristic algorithms for all instances, we use two indicators: the gap and the ranking. For a given heuristic H and a given instance, the gap is determined by using

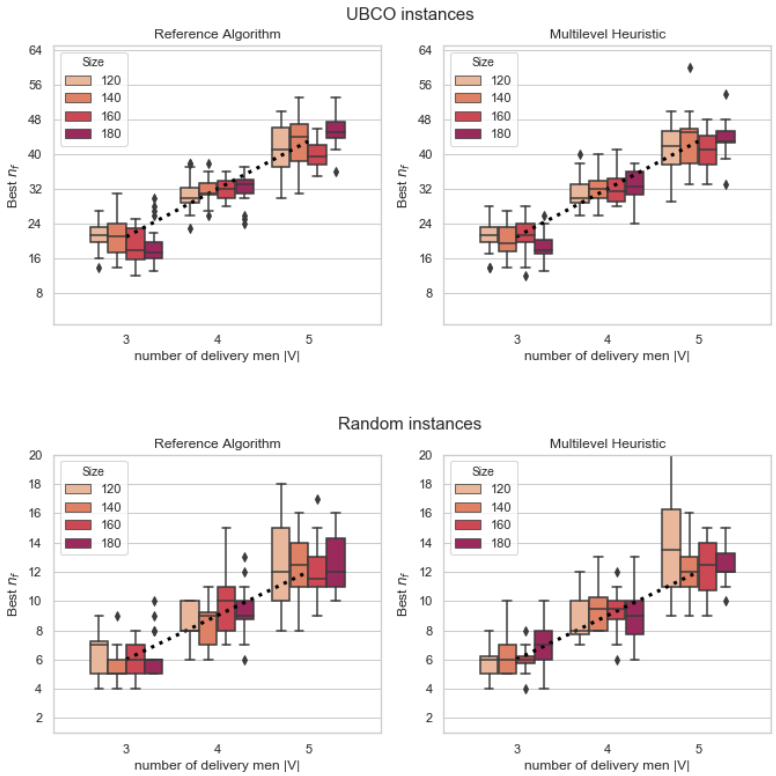

Fig. 8 Box plots of best values of n_f for RA and the ML heuristic as a function of number $|V|$ of delivery persons for UBCO and random instances and for different instance sizes. For each value of $|V|$, the box plots are sorted from left to right in increasing order of instance size.

$$\text{Gap}(H) = 100\left(\frac{z(H) - z^*}{z^*}\right), \tag{6}$$

where z^* is the value of the objective function of the best solution found by all heuristics: RA, ML, BGD, and GRASP with five iterations and ten iterations, and $z(H)$ is the value of the objective function for the best solution found by heuristic H.

The ranking indicator gives the ranking between these heuristics for each instance. The rank of 1 is assigned to the heuristic that returns the best solution and the rank of 5 is assigned to the heuristic that returns the worst solution. Several preliminary experiments were done to obtain the best values for the parameters of the BGD and the GRASP, with the results listed below:

- the learning rates for the BGD and the GRASP are $\alpha_{\text{main}} = \alpha_{\text{sec}} = 0.8$;
- the percent error for the GRASP is $\theta = 20\%$;
- the number of iterations of the GRASP is set to 5 or 10.

Table 3 shows the average gap for each heuristic on all UBCO instances and random instances. Note that the best heuristic is clearly the GRASP with five or ten iterations. Although the BGD heuristic outperforms the RA and ML methods, it performs worse than the GRASP. Finally, the RA seems to give the worst results of all heuristics.

	Average gap (%)	
Heuristic	UBCO	Random
RA	63.16	26.53
ML	48.80	25.78
BGD	14.17	6.73
GRASP 5	2.74	1.32
GRASP 10	0.26	0.03

Table 3 Average gap per instance type.

Figure 9 shows four boxplot graphics that give the gap values for each heuristic, depending on the instance type, the number $|V|$ of delivery persons, and the number of chemotherapy drugs (Size). The gap variation of each heuristic is related to the size of the boxes. The variation in the gap for the ML and RA heuristics increases when the instance size or the number of delivery persons decreases, which shows that the efficiency of these two heuristics varies strongly for small instances. Furthermore, even if the average gap for the ML is less than that for the RA, the average difference in the gap between the ML and the RA is not significant compared with the difference in the gap between the ML and the BGD. This is especially true for random instances: the average difference in the gap between the RA and ML is less than 1%, and the RA is more efficient than the ML for two instance sets (instance size $\{160, 180\}$, and $|V| = 3$). Note that the time windows to schedule and deliver jobs between the release dates and the delivery due dates are larger in the case of random instances. Therefore random instances provide more flexibility for the scheduling part and give more importance to the *Routing* phase for the solution quality. Appendix B gives detailed results for objective functions in the form of four boxplot graphics that represent the mean tardiness for each heuristic as a function of number $|V|$ of delivery persons, instance type, and number of chemotherapy drugs (Size).

To better compare the performance of the methods, we also study the percent ranking for each method (i.e., the number of times that a method gives the best result, second-best result, etc.). Figure 10 shows the percent ranks for each method and for all instances of each type. The same trend appears as in the previous results; namely, GRASP with ten iterations dominates all others heuristics, GRASP with five iterations dominates BDG heuristics that dominates ML heuristics, and the worst heuristic is the RA. Given that the distance function [Eq. (4)] was chosen to be effective for UBCO instances (Robbes et al. 2019), the ML heuristic therefore gives worse results for the random instances.

Table 4 lists the average computation times of each heuristic according to the instance type. Recall that an objective is to obtain good solutions within a reasonable computation time to ensure that we have a decision-making tool with an online algorithm that can compute a new solution every time a new event occurs. To respect

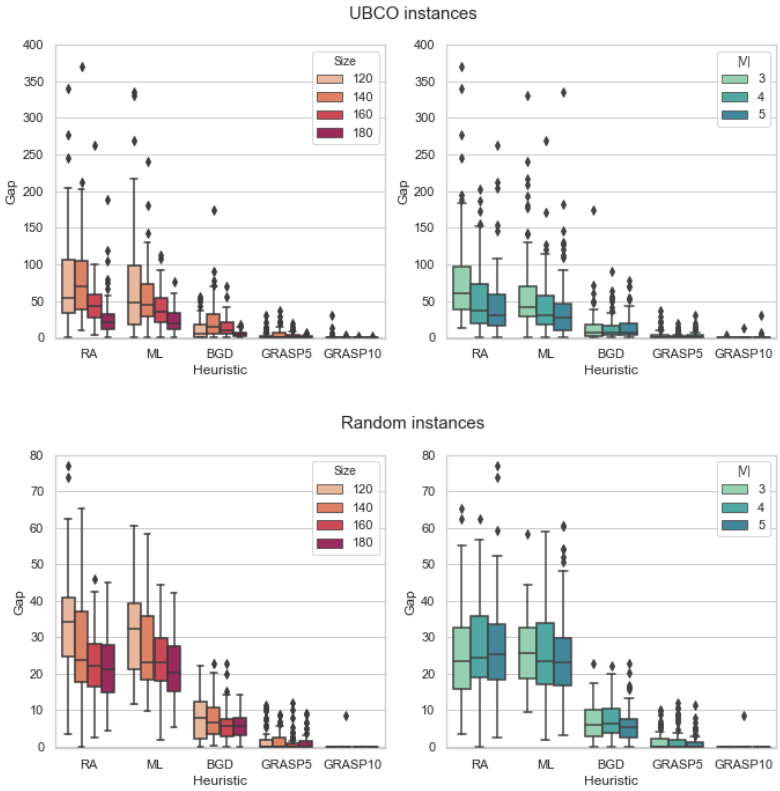

Fig. 9 Box plots of the heuristics gap (%) as a function of instances types and for different instance sizes and number $|V|$ of delivery persons. For each heuristic, the box plots are sorted from left to right in increasing order of the instance size or number $|V|$ of delivery persons.

this constraint, we fix the maximum acceptable computation time to one minute. Appendix C gives detailed results on final values of the objective function in the form of four boxplot graphics that represent the computation times for each heuristic as a function of instance type and for different numbers $|V|$ of delivery persons and numbers of chemotherapy drugs (Size).

Computation time

	UBCO		Random	
Heuristic	Average	Max.	Average	Max.
RA	0.18	0.36	0.18	0.37
ML	0.34	0.63	0.35	0.60
BGD	4.69	13.00	3.09	7.22
GRASP 5	28.07	56.98	18.50	35.52
GRASP 10	51.43	106.29	33.89	65.11

Table 4 Average and maximum computation time (s) of each heuristic for different instance types.

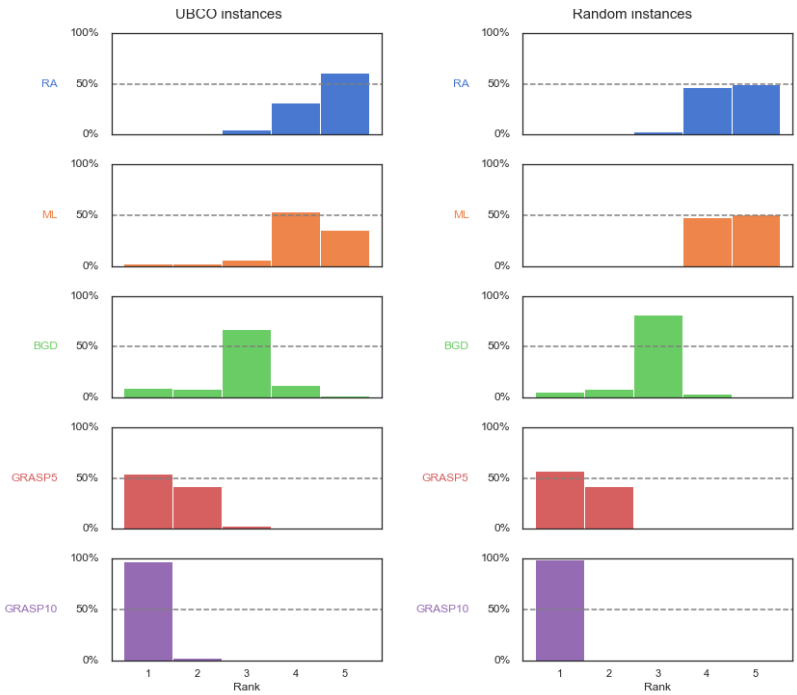

Fig. 10 Histogram of rankings of each heuristic for UBCO and random instances. In the event of a tie, the minimum rank is retained for both.

All heuristics have an average computation time of less than 1 minute. However, the maximum computation time of GRASP with ten iterations is almost 2 minutes. As expected, the RA is the fastest heuristic, and the ML heuristic is the second fastest. The computation times for the ML heuristic are roughly twice those of the RA. The time complexity of these two algorithms is mainly due to the `AH_Clustering` algorithm, which is used twice in the ML heuristic and only once in the RA. The execution of BGD algorithm takes less than 5 s, and the computation time of the GRASP algorithm depends on the number of iterations (it takes 4 to 5 s per iteration). The next section discusses the compromise between the quality of the best solution obtained and the number of iterations of the GRASP.

4.3 GRASP efficiency

The results discussed above show that the GRASP with five or ten iterations provides the best results for both random and UBCO instances. As expected, the number of iterations affects not only the quality of the best solution but also the computation time.

Figure 11 shows the gap [computed using Eq. (6)] and the computation time as a function of the number of iterations. The evolution of the computation times seems to be proportional to the number of iterations for random and UBCO instances.

The computation time is around 7 (3) s per iteration for instances with 180 (120) chemotherapy drugs. The results for the UBCO instances suggest that five iterations suffice because, in most cases, further iterations do not lead to significant improvement.

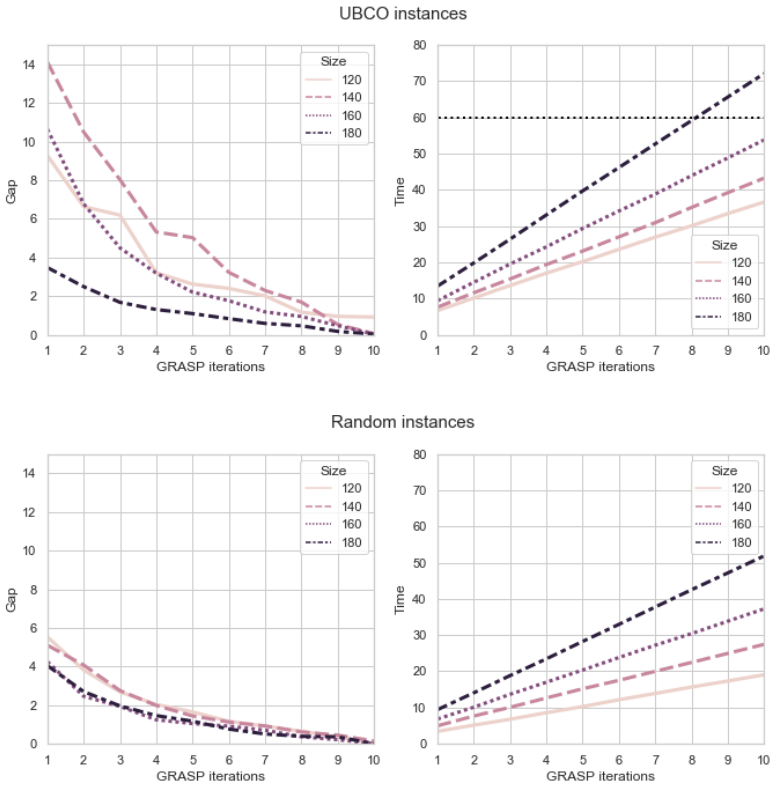

Fig. 11 GRASP efficiency as a function of number of iterations and for random and UBCO instances and different instance sizes. The left-hand plots show the gap and the right-hand plots show the computation time.

5 Conclusions and future works

This paper presents a study of a real case of chemotherapy production and delivery and proposes a model that integrates models of the hybrid flow shop scheduling problem and the multi-trip vehicle routing problem. The objective function is the minimization of the total tardiness of chemotherapy delivery. To solve the problem within minimal computation time, several heuristic algorithms are proposed. These algorithms are then applied to random-real-life instances and random instances and

the results are compared. The numerical experiments show the efficiency of the methods that schedule the production of the jobs on the basis of expected delivery batches defined by a clustering algorithm before determining the delivery planning. The proposed heuristics outperform a reference algorithm that corresponds to the current planning method and obtains good solutions within reasonable computation times (less than 1 minute). The proposed heuristics can be applied in other contexts if the scheduling model is of hybrid-flow-shop type (with necessary changes made for the number of stages and resources per stage), and the distribution model is a vehicle routing problem with or without the multi-trip aspect.

Several directions can be considered for future research. A first direction is to improve delivery planning, which currently suffices for the real-life instances but which can be improved to obtain better solutions for instances with more delivery points. Using local permutations such as 2-opt or using exact methods for the routing part also seems to be an interesting research direction. Another direction consists of revising the clustering method by modifying the distance function between jobs (e.g., using weights for the dimensions) to improve the performance of the resolution method. Another direction is to optimize a new objective function based on economic considerations, which can be done by taking into account the use of perishable resources during the production phase. These perishable resources are mainly the active ingredients used to make chemotherapy drugs and are stored in small vials, where each vial has a stability time (i.e., lifetime after opening) and a cost that can amount to thousands of euros. To avoid the high costs induced by the waste of perishable resources, production management should take into account the use of these vials. This line of research could make a significant economic impact without degrading patient care or reducing the quality of the treatment of their disease.

Acknowledgements This study is funded by the Région Centre-Val de Loire (France).

References

F. R. Abyaneh and S. Gholami. A comparison of algorithms for minimizing the sum of earliness and tardiness in hybrid flow-shop scheduling problem with unrelated parallel machines and sequence-dependent setup times. Journal of Industrial and Systems Engineering, 8(2):67–85, 2015.

M. Belo-Filho, P. Amorim, and B. Almada-Lobo. An adaptive large neighbourhood search for the operational integrated production and distribution problem of perishable products. International Journal of Production Research, 53(20):6040–6058, 2015.

D. Cattaruzza, N. Absi, and D. Feillet. The multi-trip vehicle routing problem with time windows and release dates. Transportation Science, 50(2):676–693, 2016.

S. Chahed, E. Marcon, E. Sahin, D. Feillet, and Y. Dallery. Exploring new operational research opportunities within the home care context: the chemotherapy at home. Health Care Management Science, 12 (2):179–191, 2009.

Z. Chekoubi, W. Trabelsi, and N. Sauer. The integrated production-inventory-routing problem in the context of reverse logistics: The case of collecting and remanufacturing of eol products. In 2018 4th International Conference on Optimization and Applications (ICOA), pages 1–6. IEEE, 2018.

A. Cheref, C. Artigues, J.-C. Billaut, and S. U. Ngueveu. Integrated production scheduling and delivery routing: complexity results and column generation. In International Symposium on Combinatorial Optimization, pages 439–450. Springer, 2016.

P. Devapriya, W. Ferrell, and N. Geismar. Integrated production and distribution scheduling with a perishable product. European Journal of Operational Research, 259(3):906–916, 2017.

L.-L. Fu, M. A. Aloulou, and C. Triki. Integrated production scheduling and vehicle routing problem with job splitting and delivery time windows. International Journal of Production Research, 55(20): 5942–5957, 2017.

M. Ganji, H. Kazemipoor, S. M. H. Molana, and S. M. Sajadi. A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. Journal of Cleaner Production, page 120824, 2020.

A. Gharaei and F. Jolai. A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain. Applied Soft Computing, 65:577–589, 2018.

N. Jamili, M. Ranjbar, and M. Salari. A bi-objective model for integrated scheduling of production and distribution in a supply chain with order release date restrictions. Journal of Manufacturing Systems, 40:105–118, 2016.

İ. Karaoğlan and S. E. Kesen. The coordinated production and transportation scheduling problem with a time-sensitive product: a branch-and-cut algorithm. International Journal of Production Research, 55 (2):536–557, 2017.

A. Karoonsoontawong, P. Punyim, W. Nueangnitnaraporn, and V. Ratanavaraha. Multi-trip time-dependent vehicle routing problem with soft time windows and overtime constraints. Networks and Spatial Economics, pages 1–50, 2020.

Y. Kergosien, J.-F. Tournamille, B. Laurence, and J.-C. Billaut. Planning and tracking chemotherapy production for cancer treatment: A performing and integrated solution. International Journal of Medical Informatics, 80(9):655–662, 2011.

Y. Kergosien, M. Gendreau, and J.-C. Billaut. A benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints. European Journal of Operational Research, 262(1):287–298, 2017.

S. E. Kesen and T. Bektaş. Integrated production scheduling and distribution planning with time windows. In Lean and Green Supply Chain Management, pages 231–252. Springer, 2019.

P. Lacomme, A. Moukrim, A. Quilliot, and M. Vinot. Supply chain optimisation with both production and transportation integration: multiple vehicles for a single perishable product. International Journal of Production Research, 56(12):4313–4336, 2018.

J. Lee, B.-I. Kim, A. L. Johnson, and K. Lee. The nuclear medicine production and delivery problem. European Journal of Operational Research, 236(2):461–472, 2014.

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

W. Li and W. G. Ferrell. Integrated production and distribution schedule problem with a perishable product. In IIE Annual Conference. Proceedings, page 1. Institute of Industrial and Systems Engineers (IISE), 2011.

F. Liberatore, G. Righini, and M. Salani. A column generation algorithm for the vehicle routing problem with soft time windows. 4OR, 9(1):49–82, 2011.

L. Liu and S. Liu. Integrated production and distribution problem of perishable products with a minimum total order weighted delivery time. Mathematics, 8(2):146, 2020.

L. Liu, W. Li, K. Li, and X. Zou. A coordinated production and transportation scheduling problem with minimum sum of order delivery times. Journal of Heuristics, 26(1):33–58, 2020.

C. Low, C.-M. Chang, and B.-Y. Gao. Integration of production scheduling and delivery in two echelon supply chain. International Journal of Systems Science: Operations & Logistics, 4(2):122–134, 2017.

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

A. Mazier, J.-C. Billaut, and J.-F. Tournamille. Scheduling preparation of doses for a chemotherapy service. Annals of Operations Research, 178(1):145–154, 2010.

P. L. Miranda, R. Morabito, and D. Ferreira. Optimization model for a production, inventory, distribution and routing problem in small furniture companies. Top, 26(1):30–67, 2018.

P. L. Miranda, R. Morabito, and D. Ferreira. Mixed integer formulations for a coupled lot-scheduling and vehicle routing problem in furniture settings. INFOR: Information Systems and Operational Research, 57(4):563–596, 2019.

S. Mohammadi, S. M. Al-e Hashem, and Y. Rekik. An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company. International Journal of Production Economics, 219:347–359, 2020.

S. Moons, K. Ramaekers, A. Caris, and Y. Arda. Integrating production scheduling and vehicle routing decisions at the operational decision level: a review and discussion. Computers & Industrial Engineering, 104:224–245, 2017.

F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1):86–97, 2012.

M. G. Resende and C. C. Ribeiro. Grasp: Greedy randomized adaptive search procedures. In Search Methodologies, pages 287–312. Springer, 2014.

A. Robbes, Y. Kergosien, and J.-C. Billaut. Multi-level heuristic to optimize the chemotherapy production and delivery. In International Conference on Human-Centred Software Engineering, pages 263–273. Springer, 2019.

D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, II. An analysis of several heuristics for the traveling salesman problem. SIAM Journal on Computing, 6(3):563–581, 1977.

S. Ruder. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016.

R. Ruiz and J. A. Vázquez-Rodríguez. The hybrid flow shop scheduling problem. European Journal of Operational Research, 205(1):1–18, 2010.

M. Savelli, M. Roche, C. Curti, C. Bornet, P. Rathelot, M. Montana, and P. Vanelle. Methods to control anticancer chemotherapy preparations ranked by risk analysis. Die Pharmazie-An International Journal of Pharmaceutical Sciences, 73(5):251–259, 2018.

B. C. Shelbourne, M. Battarra, and C. N. Potts. The vehicle routing problem with release and due dates. INFORMS Journal on Computing, 29(4):705–723, 2017.

J. A. Snyman. Practical Mathematical Optimization. Springer, 2005.

M. Tamannaei and M. Rasti-Barzoki. Mathematical programming and solution approaches for minimizing tardiness and transportation costs in the supply chain scheduling problem. Computers & Industrial Engineering, 127:643–656, 2019.

R. F. Tavares-Neto and M. S. Nagano. An iterated greedy approach to integrate production by multiple parallel machines and distribution by a single capacitated vehicle. Swarm and Evolutionary Computation, 44:612–621, 2019.

Ö. Tosun, M. Marichelvam, and N. Tosun. A literature review on hybrid flow shop scheduling. International Journal of Advanced Operations Management, 12(2):156–194, 2020.

C. A. Ullrich. Integrated machine scheduling and vehicle routing with time windows. European Journal of Operational Research, 227(1):152–165, 2013.

D. Wang, J. Zhu, X. Wei, T. Cheng, Y. Yin, and Y. Wang. Integrated production and multiple trips vehicle routing with time windows and uncertain travel times. Computers & Operations Research, 103:1–12, 2019a.

J. Wang, S. Yao, J. Sheng, and H. Yang. Minimizing total carbon emissions in an integrated machine scheduling and vehicle routing problem. Journal of Cleaner Production, 229:1004–1017, 2019b.

S. Wang, R. Wu, F. Chu, and J. Yu. Variable neighborhood search-based methods for integrated hybrid flow shop scheduling with distribution. Soft Computing, pages 1–20, 2019c.

E. Yağmur and S. E. Kesen. A memetic algorithm for joint production and distribution scheduling with due dates. Computers & Industrial Engineering, 142(106342), 2020.

X. Zou, L. Liu, K. Li, and W. Li. A coordinated algorithm for integrated production scheduling and vehicle routing problem. International Journal of Production Research, 56(15):5005–5024, 2018.

Appendix

A Mathematical formulation

Table 5 lists the input parameters of the problem and Table 6 presents the variables of the mathematical formulation.

Notation	Description
$\mathscr{J}$	set of jobs
$\mathscr{I}$	set of isolators
K	set of `Sterilization` batches
M_i	set of operators connected to isolator I_i
O	set of all operators ($O = \bigcup_{I_i \in \mathscr{I}} M_i$)
V	set of delivery persons
T	set of trips
Q_i	sterilizer capacity (number of jobs) of isolator I_i
p^S	`Sterilization` processing time
p^A	`Control` processing time
$r_j, \forall J_j \in \mathscr{J}$	release date of J_j
$p_j^O, \forall J_j \in \mathscr{J}$	`Preparation` processing time of J_j
$d_j, \forall J_j \in \mathscr{J}$	due date of J_j
$u_j, \forall J_j \in \mathscr{J}$	oncology unit where job J_j has to be delivered
$tt_{j,j'}, \forall J_j \in \mathscr{J}\ \forall J_{j'} \in \mathscr{J}$	travel time between u_j and $u_{j'}$

Table 5 Input parameters

Notation and domain	Description
	`Sterilization`
$x_j^{o,k} \in \{0,1\}, \forall J_j \in \mathscr{J}, \forall o \in O, \forall k \in K$	$x_j^{o,k} = 1$ if job J_j is sterilized in the kth sterilization batch associated to the operator o and is prepared by operator o, 0 otherwise.
	`Preparation`
$y_j^o \in \{0,1\}, \forall J_j \in \mathscr{J}, \forall o \in O$	$y_j^o = 1$ if job J_j is prepared by the operator o, 0 otherwise.
$\gamma_{j,j'} \in \{0,1\}, \forall J_j \in \mathscr{J}\ \forall J_{j'} \in \mathscr{J}$	$\gamma_{j,j'} = 1$ if job J_j is prepared before job $J_{j'}$ by the same operator, 0 otherwise.
$C_j^O \in \mathbb{R}_+, \forall J_j \in \mathscr{J}$	`Preparation` completion time of J_j.
	`Control`
$\delta_{j,j'} \in \{0,1\}, \forall J_j \in \mathscr{J}, \forall J_{j'} \in \mathscr{J}$	$\delta_{j,j'} = 1$ if job J_j is controlled before job $J_{j'}$, 0 otherwise.
$C_j^A \in \mathbb{R}_+, \forall J_j \in \mathscr{J}$	`Control` completion time of J_j
	`Delivery`
$z_j^{v,t} \in \{0,1\}, \forall J_j \in \mathscr{J}, \forall v \in V, \forall t \in T$	$z_j^{v,t} = 1$ if job J_j is delivered by the delivery person v during its tth trip, 0 otherwise.
$\bar{z}^{v,t} \in \{0,1\}, \forall v \in V, \forall t \in T$	$\bar{z}^{v,t} = 1$ the trip t of the delivery person v is not empty and delivers at least one job, 0 otherwise.
$\zeta_{j,j'} \in \{0,1\}, \forall J_j \in \mathscr{J}\ \forall J_{j'} \in \mathscr{J}$	$\zeta_{j,j'} = 1$ if a delivery person travels from unit u_j to unit $u_{j'}$, 0 otherwise.
$\alpha^{v,t} \in \mathbb{R}_+, \forall v \in V, \forall t \in T$	Departure date for trip tth of the delivery person v.
$D_j \in \mathbb{R}_+, \forall J_j \in \mathscr{J}$	Delivery date of J_j.
$T_j \in \mathbb{R}_+, \forall J_j \in \mathscr{J}$	Tardiness of J_j.

Table 6 Decision variables

The mathematical programming formulation is given below. Note that the formulation is not linear, although it is possible to linearize the model by using big-M constraints and additional variables. We assume that the distance matrix $\left(tt_{j,j'}\right)$ satisfies the triangle inequality. The objective function is to minimize the total tardiness of delivery:

$$\text{MIN} \sum_{J_j \in \mathscr{J}} T_j \tag{7}$$

Sterilization capacity is not exceeded:

$$\sum_{o \in M_i} \sum_{J_j \in \mathscr{J}} x_j^{o,k} \leq Q_i \quad \forall I_i \in \mathscr{I},\ \forall k \in K \tag{8}$$

The job J_j is sterilized by the corresponding sterilizer of the operator o:

$$\sum_{k \in K} x_j^{o,k} = y_j^o \quad \forall J_j \in \mathscr{J},\ \forall o \in O \tag{9}$$

The `Preparation` of job J_j proceeds after its `Sterilization`:

$$kp^S x_j^{o,k} + p_j^O \leq C_j^O \quad \forall J_j \in \mathscr{J},\ \forall o \in O,\ \forall k \in K \tag{10}$$

Job J_j is prepared once:

$$\sum_{o \in O} y_j^o = 1 \quad \forall J_j \in \mathscr{J} \tag{11}$$

The `Preparation` of job J_j starts after its release date r_j:

$$r_j + p_j^O \leq C_j^O \quad \forall J_j \in \mathscr{J} \tag{12}$$

The disjunctive constraint ensures that no two jobs can be scheduled on the same operator at the same time:

$$C_j^O \gamma_{j,j'} + p_{j'}^O \leq C_{j'}^O \quad \forall J_j \in \mathscr{J},\ \forall J_{j'} \in \mathscr{J},\ \textit{and}\ J_j \neq J_{j'} \tag{13}$$

A precedence constraint exists between two jobs if they are prepared by the same operator:

$$\gamma_{j,j'} + \gamma_{j',j} \geq y_j^o + y_{j'}^o - 1 \quad \forall o \in O,\ \forall J_j \in \mathscr{J},\ \forall J_{j'} \in \mathscr{J},\ \textit{and}\ J_j \neq J_{j'} \tag{14}$$

The `Control` of job J_j begins after its `Preparation` completion time C_j^O:

$$C_j^O + p^A \leq C_j^A \quad \forall J_j \in \mathscr{J} \tag{15}$$

The precedence of jobs is one-way:

$$\delta_{j,j'} + \delta_{j',j} = 1 \quad \forall J_j \in \mathscr{J},\ \forall J_{j'} \in \mathscr{J},\ \textit{and}\ J_j \neq J_{j'} \tag{16}$$

Only one job is analyzed at a time:

$$C_j^A \delta_{j,j'} + p^A \leq C_{j'}^A \quad \forall J_j \in \mathscr{J},\ \forall J_{j'} \in \mathscr{J},\ \textit{and}\ J_j \neq J_{j'} \tag{17}$$

The starting time of trip t for the delivery person v starts after the `Control` completion time of each job in the delivery batch:

$$C_j^A z_j^{v,t} \leq \alpha^{v,t} \quad \forall J_j \in \mathscr{J},\ \forall v \in V,\ \forall t \in T \tag{18}$$

Job J_j is delivered once:

$$\sum_{v \in V} \sum_{t \in T} z_j^{v,t} = 1 \quad \forall J_j \in \mathscr{J} \tag{19}$$

A noempty trip is composed of at least one job:

$$\sum_{J_j \in \mathscr{J}} z_j^{v,t} \leq |\mathscr{J}| \bar{z}^{v,t} \quad \forall v \in V,\ \forall t \in T \tag{20}$$

There is only one departure for each no empty trip:

$$\sum_{J_j \in \mathscr{J}} \zeta_{0,j} z_j^{v,t} = \bar{z}^{v,t} \quad \forall v \in V,\ \forall t \in T \tag{21}$$

The delivery date at each location is bound by the trip starting date:

$$\sum_{v \in V} \sum_{t \in T} \alpha^{v,t} z_j^{v,t} + tt_{0,j} \leq D_j \quad \forall J_j \in \mathscr{J} \tag{22}$$

Each location is visited only once (input flow constraint):

$$\sum_{J_{j'} \in \mathscr{J} \setminus \{J_j\}} \zeta_{j',j} z_{j'}^{v,t} + \zeta_{0,j} z_j^{v,t} = z_j^{v,t} \quad \forall J_j \in \mathscr{J},\ \forall v \in V,\ \forall t \in T \tag{23}$$

Each location is left only once (output flow constraint):

$$\sum_{J_{j'} \in \mathscr{J} \setminus \{J_j\}} \zeta_{j,j'} z_{j'}^{v,t} + \zeta_{j,0} z_j^{v,t} = z_j^{v,t} \quad \forall J_j \in \mathscr{J},\ \forall v \in V,\ \forall t \in T \tag{24}$$

The delivery date depends of the trip sequence:

$$(D_j + tt_{j,j'}) \zeta_{j,j'} \leq D_{j'} \quad \forall J_j \in \mathscr{J},\ \forall J_{j'} \in \mathscr{J},\ \textit{and}\ J_j \neq J_{j'} \tag{25}$$

Each no empty trip terminates at the depot:

$$\sum_{J_j \in \mathscr{J}} \zeta_{j,0} z_j^{v,t} = \bar{z}^{v,t} \quad \forall v \in V,\ \forall t \in T \tag{26}$$

A trip can start only after the delivery person returns his or her preceding trip:

$$(D_j + tt_{j,0}) z_j^{v,t'} \leq \alpha^{v,t} \quad \forall J_j \in \mathscr{J},\ \forall v \in V,\ \forall t \in T, \forall t' \in \{1, ..., t-1\} \tag{27}$$

Tardiness is computed from the delivery date and the due date:

$$T_j \geq D_j - d_j \quad \forall J_j \in \mathscr{J} \tag{28}$$

B Results of mean tardiness

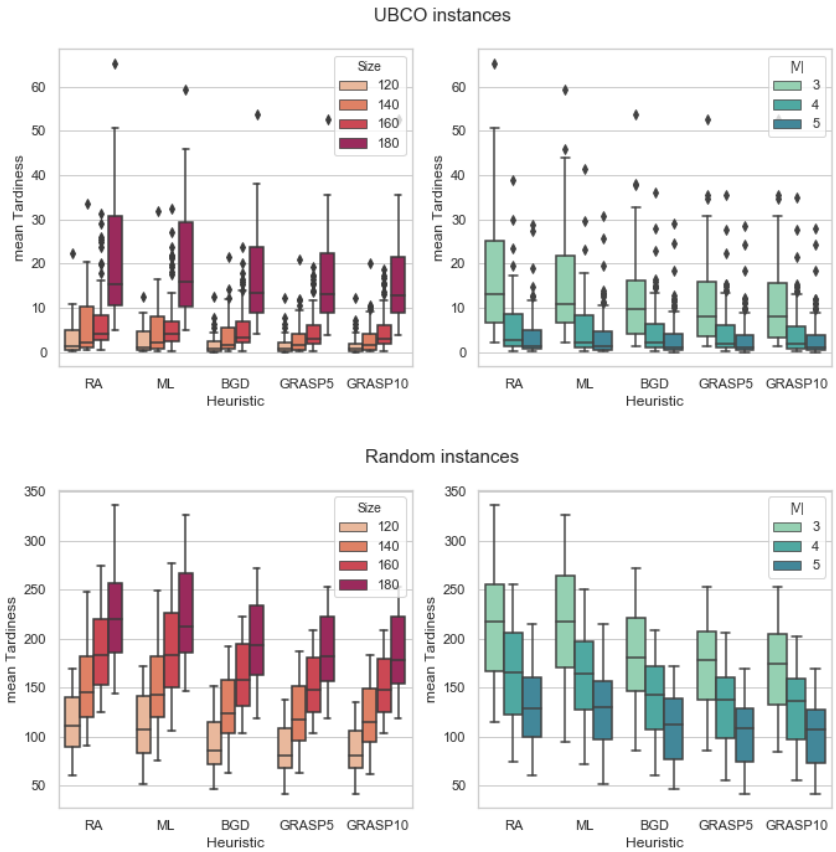

Fig. 12 Boxplots of mean tardiness ($\frac{T_j}{|\mathscr{J}|}$) as a function heuristic and for different instance types, instance sizes, and numbers of delivery persons. For each heuristic, the boxplots are sorted from left to right in increasing order of instance size or number $|V|$ of delivery persons.

C Computation times of heuristics

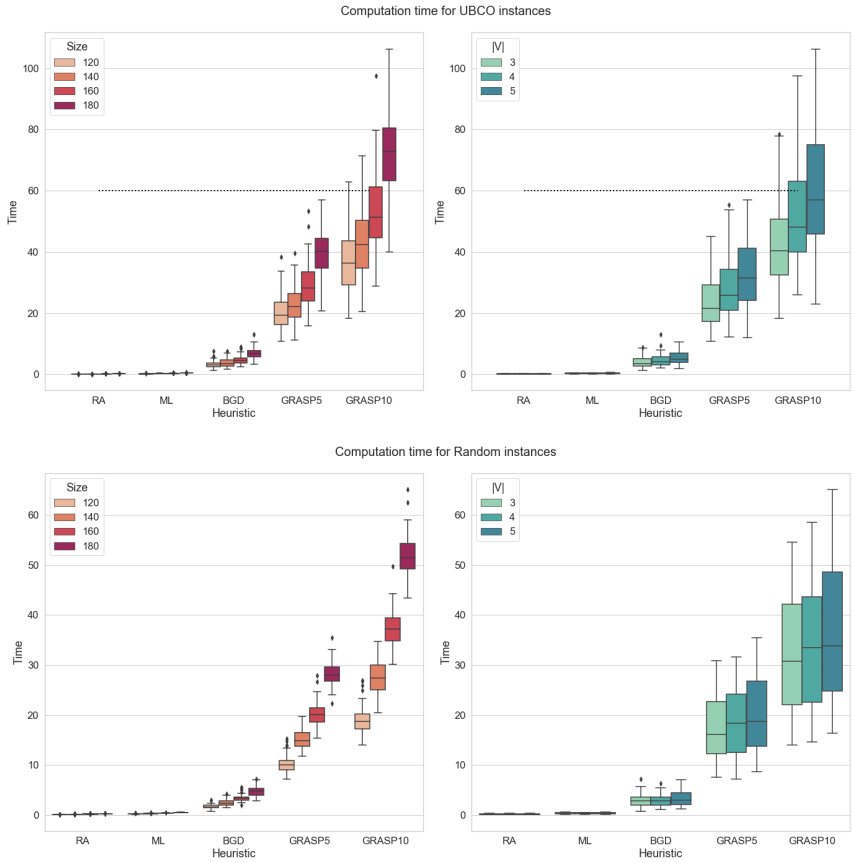

Fig. 13 Boxplot of computation time as a function of heuristic and for different instance types, instance sizes, and numbers of delivery persons. For each heuristic, the boxplots are sorted from left to right in increasing order of instances size or number $|V|$ of delivery persons.