



**HAL**  
open science

# Home Chemotherapy Planning: An Integrated Production Scheduling and Multi-Trip Vehicle Routing Problem

Yasemin Arda, Diego Cattaruzza, Véronique François, Maxime Ogier

► **To cite this version:**

Yasemin Arda, Diego Cattaruzza, Véronique François, Maxime Ogier. Home Chemotherapy Planning: An Integrated Production Scheduling and Multi-Trip Vehicle Routing Problem. 2022. hal-03728086

**HAL Id: hal-03728086**

**<https://hal.science/hal-03728086v1>**

Preprint submitted on 20 Jul 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.

# Home Chemotherapy Planning: An Integrated Production Scheduling and Multi-Trip Vehicle Routing Problem

Yasemin Arda, Diego Cattaruzza, Véronique François, Maxime Ogier

## Abstract

Home chemotherapy systems allow the administration of cancer treatments at a patient's residence, avoiding an admission to inpatient care facilities. This innovative health care model is interesting both economically and on a human level. It also raises several logistical challenges. This paper focuses on one of the optimization problems arising in the context of home chemotherapy services, where a complex scheduling problem underlies the operational planning process. Indeed, some injectable chemotherapy drugs may remain stable only during a few hours after being produced. Consequently, their production has to be carefully scheduled jointly with their administration, which takes place at the patients homes during a predefined time window. This gives rise to an integrated production scheduling and vehicle routing problem, that we address using a large neighborhood search approach. Production and administration sequences are iteratively modified, while a linear program is used to determine optimal production and administration start times for the candidate sequences. We analyze the impact of the linear program and establish that it is a crucial component of the proposed method. We then provide insights about the cost of taking into consideration time-related aspects of the problem, i.e., integrated planning horizons, drug stability times, and administration time windows.

**Keywords** Production scheduling and vehicle routing problem; Multi-trip vehicle routing; Large neighborhood search; Chemotherapy planning

## 1 Introduction

In many developed countries, home chemotherapy is a rising trend. Home chemotherapy services aim to assist cancer patients to remain safe and comfortable at home

while continuing to receive their treatment, avoiding hospitalization or admission to outpatient chemotherapy facilities. Home chemotherapy also contributes to the employability of patients, enabling them to remain active for longer periods and in better health. Besides increasing the comfort of the patients, home chemotherapy may help relieve congestion in outpatient chemotherapy services. In the nineties, an Australian case study [Lowenthal et al., 1996] demonstrated that home chemotherapy was at the same time safe and cost effective. A report of the European Commission [European Commission, 2016] states that institutional care systems typically have higher costs than home care systems, which should be encouraged when the patient’s condition allows it to favor at the same time independent living and cost-effectiveness. Thus, shifting from a hospital-centered to a home-based care system can clearly be seen as a possible direction to reinforce the financial stability of health systems. For all those reasons, home chemotherapy sparks the interest of healthcare authorities, as does hospital at home in general.

At the operational level, a complex scheduling problem underlies the daily home chemotherapy process. It calls for the determination of an integrated drug production and administration schedule. Indeed, injectable preparations for cancer treatment sometimes have a short stability time, i.e., they may expire within a few hours after their production start time. Consequently, they may not be produced ahead of time and then stored before being administered. The resulting absence of inventories implies that the production of drugs has to be carefully scheduled jointly with their administration. Moreover, when chronotherapy is used in cancer treatment, the timing of drug administration takes into account biological rhythms of patients. This may result in a short recommended time window to administer a given drug. Since the patients comfort is essential for a quick recovery, their preferences need also to be taken into account to determine the time window of the treatment administration. Doing so also allows patients to remain professionally active when possible. For all the above mentioned reasons, establishing a schedule for the production and administration of home chemotherapy drugs is a complex task which may be a turn-off for medical teams. If handled properly, however, it may contribute to improve the quality of service for the patients and to relieve the work of the medical staff inside the hospital.

The operational problem described in this work stems from an envisioned home chemotherapy system based on the current emerging practice observed in several Belgian and French healthcare institutions. Our first contribution is to propose a solution method based on the strong interdependence of the vehicle routing and production scheduling subproblems. A linear program, which usefulness is clearly demonstrated through computational experiments, is embedded in a large neighborhood search framework and several algorithmic components specifically address the temporal connection between the two subproblems. We propose a set

of test instances that encompass realistic features of home chemotherapy systems. Since several algorithmic parameters are required, we use the `irace` package to configure the proposed solution method. As a second contribution, we provide insights about the impact of several instance parameters, and about their influence on the integration of the two subproblems. In particular, we analyze the cost of integration, i.e., we investigate to which extent the absence of inventories due to short stability times increases the cost of the routing and production schedules.

The remainder of this paper is organized as follows. Section 2 provides a description of the home chemotherapy planning problem under consideration. A literature review is then given in Section 3. Section 4 presents the proposed solution approach, which combines large neighborhood search (LNS) with mathematical programming operators. Section 5 provides numerical results. Conclusions and some prospects are drawn in Section 6.

## 2 Problem description

In the considered problem, a set of patients must receive their chemotherapy treatment at home on a given day. A single personalized drug must be produced for each patient. In the following, we consider a set  $\mathcal{J}$  of patients, where each patient is associated with two operations: a drug production operation and a drug administration operation. Thus, each patient  $i \in \mathcal{J}$  is characterized by:

- a processing time  $P_i$ , which is the time needed to produce the personalized drug at the hospital pharmacy,
- a stability time  $ST_i$ , which denotes the maximum delay between the start time of the production operation and the start time of the administration operation,
- an administration time  $S_i$ , which is the time needed to administer the drug at the patient's home,
- an administration time window  $[E_i, L_i]$ , within which the administration operation can start.

The drugs are produced at the hospital pharmacy by a set  $\mathcal{P}$  of pharmacists. Each drug must be processed without preemption by a single pharmacist and each pharmacist can produce any one drug at a time. We assume that pharmacists have homogeneous production skills, i.e., the processing time  $P_i$  of a given drug  $i$  does not depend on the pharmacist that produces the drug. Given these assumptions, we model pharmacists as identical parallel machines and drugs as jobs composed of a single operation analogously to the production scheduling literature.

The drug administration planning can be modelled as a multi-trip vehicle routing problem. A set  $\mathcal{N}$  of specialized nurses administer the drugs to patients at home. Each nurse performs an administration journey that starts and ends at the hospital, denoted by 0. During the journey, recently produced drugs may be loaded at the hospital between patient visits. A set of consecutive patient visits that starts and ends at the hospital forms a trip. The time required for loading operations at the hospital,  $S_0$ , is supposed constant for each trip, i.e., independent of the specific subset of drugs to be loaded. Thus, without loss of generality, we assume  $S_0 = 0$ . The traveling time between two patients  $i$  and  $j$  is  $T_{ij}$ , while  $T_{0i}$  (resp.  $T_{i0}$ ) denotes the traveling time between the hospital and patient  $i$  (resp. between patient  $i$  and the hospital). Each patient  $i$  must be visited within a given hard time window  $[E_i, L_i]$ . A nurse arriving early has to wait, and being late is prohibited.

Pharmacists and nurses have a limited working duration  $D^{max}$ . They must start and end their working shift within the planning horizon, defined by the time window  $[E_0, L_0]$ . We suppose, without loss of generality, that  $E_0 = 0$ .

The problem calls for the determination of a complete production and administration schedule while ensuring that the drug administration for each patient starts during the corresponding time window, each drug is stable at the time of its administration, the production process of each drug is completed before the start time of the trip to which the corresponding patient is assigned, and the working duration of none of the pharmacists and nurses exceeds its maximum limit. Under those constraints, the following decisions have to be taken:

- the production of each drug must be assigned to a pharmacist,
- a production schedule has to be determined for each pharmacist,
- the administration of each drug must be assigned to a nurse,
- a routing schedule composed of one or more trips that do not overlap in time has to be determined for each nurse.

Consequently, the problem under consideration includes two integrated sub-problems: a production scheduling problem with identical parallel machines, and a multi-trip vehicle routing problem with time windows. The two subproblems are strongly integrated because of 1) stability constraints, i.e., after the effective start of its production process, each drug administration should start before the stability time expires, and 2) trip start time constraints, i.e., the production of all drugs to be administered during a given trip should be completed before the start time of this trip.

Let  $e_p^P$  be the start time and  $l_p^P$  the end time of the working shift of pharmacist  $p \in \mathcal{P}$ . The shift start time  $e_p^P$  is determined by the start time of the first

production operation executed by pharmacist  $p$ , while the shift end time  $l_p^P$  is the completion time of the last production operation. Similarly, let  $e_n^R$  be the start time and  $l_n^R$  the end time of the working shift of nurse  $n \in \mathcal{N}$ , where  $e_n^R$  is the start time of the first trip and  $l_n^R$  the completion time of the last trip performed by nurse  $n$ . The objective is to minimize the total working duration of the nurses and pharmacists:  $\sum_{p \in \mathcal{P}} (l_p^P - e_p^P) + \sum_{n \in \mathcal{N}} (l_n^R - e_n^R)$ .

The total working duration takes into account the drug production and administration times, the traveling and waiting times of nurses, and the idle times of pharmacists. Such an objective function is very relevant since pharmacists and nurses are highly skilled and may execute other value added operations at the hospital before or after their assigned home chemotherapy operations. Keeping them idle may increase significantly the opportunity costs of the system.

### 3 Literature Review

The problem we consider relates to well-studied fields of operations research: production scheduling and vehicle routing. In practice, even if the two subproblems are strongly interconnected, they are often solved sequentially. The production schedule is established first, serving as a basis to compute the routing plan. If the production schedule is known, then the routing component of the problem considered in this work may be modeled as a nurse routing problem, which concerns routing decisions for nurses that provide care at the patients' homes. The nurse routing problem has been studied for more than 20 years [e.g., Rasmussen et al., 2012, Braekers et al., 2016]. Most nurse routing problems studied in the literature are vehicle routing problems which encompass preoccupations related to the home healthcare sector such as workload balance, patient preferences with respect to nurses, or staff qualifications. In the problem studied in this work, the drug availability depends on its production completion time, implying a release date on each drug to be administered. Since the nurses are allowed to perform multiple trips, this gives rise to a multi-trip vehicle routing problem with time windows and release dates similar to the one studied in Cattaruzza et al. [2016].

However, in the literature, in situations where keeping inventories can not be considered, several authors have obtained substantial cost decreases by integrating the optimization of the production scheduling and the vehicle routing problems instead of solving them sequentially [Moons et al., 2017]. In the following, we concentrate on studies that treat the production and the routing problems jointly.

As mentioned before, the problem we consider integrates production scheduling and vehicle routing decisions. However, it should not be mistaken for the production-routing problem (PRP) surveyed by Adulyasak et al. [2015]. Indeed, the PRP combines lot-sizing and routing decisions on a multi-period basis. On

the production side, the question is not to determine a precise production schedule for one period, but rather to decide in which periods to produce. Since multiple periods are considered, inventories may be built up in the PRP. On the contrary, our work falls into a different category, surveyed by Chen [2010] and more recently by Moons et al. [2017], referred to as the production scheduling and vehicle routing problem (PS-VRP), which focuses on single-period integrated production scheduling and vehicle routing problems. In a PS-VRP, a detailed production schedule must be established in coordination with vehicle routes and delivery schedules. The related literature is quite scarce in contrast with the extreme diversity of contexts and problems. As noted by Ullrich [2013], PS-VRPs are relevant in situations where the size of inventories is too small to act as a buffer between production and distribution. Keeping inventories makes no sense when products have a short lifespan as the drugs considered in this work. Other examples may be found in several industries as hot meal, newspaper, or ready-mix concrete. For example, Garcia et al. [2004] study the distribution of concrete that must be swiftly delivered in order to avoid solidification. They consider the problem of scheduling concrete orders with a no-wait policy: orders are manufactured and immediately delivered to clients using a homogeneous fleet of vehicles. They must be served exactly at their due dates. Even for products with a longer lifespan, supply chain management strategies may imply that few or no inventories are kept between the production and the distribution activities, as it is the case in just-in-time or make-to-order contexts. For example in computer production, each customer can choose among several options for each component. As a consequence, the number of possible combinations is huge and making a stock for each final product is not sustainable for the company. Thus, computers are produced and then immediately delivered (see Chen and Vairaktarakis [2005]), in an assemble-to-order context.

There is no commonly accepted version of the PS-VRP. Different variants arose based on customer and product characteristics, and depending on the considered production and distribution environments. In our work, we consider a parallel machine environment in conjunction with a fleet of vehicles allowed to perform multiple trips as in Ullrich [2013], Lee et al. [2014], Kergosien et al. [2017], Wang et al. [2019], and Robbes et al. [2021]. Even if those five works share some similarities with ours, they are still very different in terms of problem characteristics as detailed below. Ullrich [2013] studies a PS-VRP motivated by the automotive industry, where each customer orders one or more personalized items (with no lifespan consideration). Parallel machines with machine-dependent ready times produce items that have to be delivered to customers within a given hard time window by heterogeneous vehicles. The objective is to minimize the total tardiness. The authors use a genetic algorithm to solve the integrated problem up to 100 jobs. Their numerical experiments show large improvements when solv-

ing the integrated problem compared to a decomposed approach that treats the production and the routing parts sequentially. Lee et al. [2014] consider an integrated production and delivery problem in the context of nuclear medicine, where an isotope with a 110 minutes half-life needs to be delivered to several medical end-users. Each customer orders a given quantity of isotope to be delivered within a specified time window. Production batches are launched in parallel cyclotrons at predetermined start times. On the production side, orders have to be assigned to production batches. Also, the quantities to be produced must be determined taking into account the deterioration of the isotope over time. Instances with up to 100 customers are solved using a large neighborhood search. Kergosien et al. [2017] consider a problem motivated, like ours, by the case of home chemotherapy. There are no time windows for the drug administration but an administration due date is fixed by the doctor. The objective is to minimize the maximum tardiness with respect to those drug administration due dates. The authors develop a heuristic method based on Benders decomposition to solve instances with a single vehicle and up to 40 patients. Wang et al. [2019] study a problem with machine-dependant ready times, time windows, and uncertain travel times. The objective is to minimize the travel cost, and a penalty is incurred for the cost due to tardiness. The proposed memetic algorithm allows to take into account the risk preferences of the decision maker in managing uncertainties. Numerical results are reported for instances containing up to 50 customers. Robbes et al. [2021] study the problem of producing and delivering chemotherapy drugs from a production site to different oncology units of an hospital where injections are performed. Drug production is associated with a release date corresponding to the final doctor check and validation, and with a due date that represents the desirable delivery time of the drug to the assigned unit. The production process is modelled as a three-stage hybrid flow shop scheduling problem. The objective is to minimize the total tardiness. Instances with up to 180 drugs and three delivery points are tackled with several heuristics.

In Kergosien et al. [2017], like in this work, the product has a limited lifespan and batching is not considered. Several other different PS-VRP versions have been devoted to products with a short lifespan. Most of the time, batching is used in the following way: goods to be delivered within a same trip are produced sequentially without interruption as a production batch. The product lifetime starts as the batch production starts or ends. For example, in Geismar et al. [2008], customers are served by a single truck allowed to perform multiple trips. There is a single plant, with a specified production rate, that may be considered as a single machine. Each trip delivers the products of a production batch and the lifetime of the products begins when the production batch is completed. The authors use an evolutionary algorithm to solve instances with up to 50 customers. Devapriya



et al. [2017] consider an extension where the fleet size is a decision variable. They solve instances with up to 40 customers, again using an evolutionary algorithm. Lacomme et al. [2018] propose another extension with a given fleet size and use a hybridized metaheuristic to solve large-scale instances with up to 200 customers. A branch-and-cut exact algorithm has also been proposed recently by Karaođlan and Kesen [2017] to solve the problem introduced in Geismar et al. [2008]. Chen et al. [2009] develop a model for fresh food distribution where the products deteriorate in time. A fleet of vehicles serves customers with stochastic demands. Each product starts to deteriorate at the production start time of the batch to be delivered in the same vehicle. On the contrary, in Armstrong et al. [2008] and Viergutz and Knust [2014], the end of the lifetime of each product is determined by its production start time, as it is the case in this work. However, these two works consider a problem where all perishable products must be produced before being served by a single vehicle in a single trip, and the delivery sequence of customers is known a priori. The objective is to maximize the quantity of goods delivered.

We consider that both production and routing costs are functions of the working time. Moons et al. [2017] note that production costs are not usually considered in the integrated production-routing literature, since all products need to be produced. However, in the case of home chemotherapy, pharmacists have to don a special outfit. This costs time and sterilization material, implying that pharmacists will wait between production operations in case of idle time, instead of removing their outfit to perform other tasks in the pharmacy. Thus, the potential resulting waiting time may be viewed as an opportunity cost. As for the routing costs, François et al. [2019] showed that, in presence of time windows, considering the total duration instead of the total travel time leads to much more realistic solutions, avoiding large amounts of waiting time. This is especially relevant in this case, since specialized nurses can perform duties at the hospital before or after their journey instead of waiting in the vehicle.

## 4 A large neighborhood search heuristic

To solve the problem described in Section 2, we propose to use an LNS [Shaw, 1998] that iteratively removes and reinserts drug production and administration operations to create new solutions.

### 4.1 Solution representation

A solution to the studied integrated problem is composed of a production schedule and an administration schedule. They both rely on a set of sequences for pharmacists or for nurses, respectively. In the following, the distinction between a set

of sequences and a schedule is important, because several algorithmic components modify one or more sequences without recomputing an associated schedule.

For the routing subproblem, in particular, the sequence of visits performed by a nurse is represented as a giant tour. The journey of a nurse is a sequence of operations that starts with a first loading operation at the hospital, continues with the administration operations of the assigned patients and ends with a last visit at the hospital. Additional reloading operations may occur within the sequence between administration operations. They are the trip delimiters inside the journey. The multi-trip operators detailed in François et al. [2016], which are designed to modify journeys without decomposing them into their constituting trips, are employed to remove and reinsert administration and reloading operations within the routing solutions. Using these multi-trip operators, the insertion of an administration operation may be accompanied by the insertion of a reloading operation at the hospital. When an administration operation is removed, the concerned trip may be merged together with one or both adjacent trips.

## 4.2 Search space

During the execution of the LNS algorithm, we allow visiting infeasible solutions obtained by relaxing stability, trip start time, time window, and maximum working duration constraints. In the following, we denote a solution of this relaxed version of the problem by  $\mathcal{S} = \mathcal{S}^P \cup \mathcal{S}^R$ , where  $\mathcal{S}^P$  is the corresponding solution of the production subproblem, i.e., the production schedule of  $\mathcal{S}$ , and  $\mathcal{S}^R$  is the corresponding solution of the routing subproblem, i.e., the routing schedule of  $\mathcal{S}$ .

Nagata et al. [2010] initially introduced the concept of time warp to allow late arrivals in the context of vehicle routing problems with time windows. The time warp allows, when a task is overdue in a given schedule, to fictively start the task execution at the end of its time window, i.e., just on time. The quantity of time units that are traveled back in time this way is called the time warp usage. In this work, we use the time warp concept in three different situations: *(i)* to allow violation of patients' time windows, *(ii)* to allow violation of stability times, and *(iii)* to allow producing a drug late w.r.t. the start time of the corresponding trip. When a nurse arrives late at a patient's home in case *(i)* or *(ii)*, or when a pharmacist starts a drug production late in case *(iii)*, we pretend that the concerned worker is allowed to travel back in time to start to execute the operation just on time. That is, the production and administration start times of the drugs fictively never violate the relaxed constraints due to the use of a time warp mechanism. The total time warp of a given solution  $\mathcal{S}$ , i.e., the total quantity of time units that have to be traveled back in time by all the workers, is recorded as  $\delta(\mathcal{S})$ .

In order to take into account the violations of the maximum working duration constraints, we define an overtime value for each pharmacist and each nurse. For

each pharmacist  $p \in \mathcal{P}$ , the overtime is calculated as  $\gamma_p^P = \max\{0, (l_p^P - e_p^P) - D^{max}\}$ , while for each nurse  $n \in \mathcal{N}$ , it is computed as  $\gamma_n^R = \max\{0, (l_n^R - e_n^R) - D^{max}\}$ . The total overtime of  $\mathcal{S}$  is defined as  $\gamma(\mathcal{S}) = \sum_{p \in \mathcal{P}} \gamma_p^P + \sum_{n \in \mathcal{N}} \gamma_n^R$ .

The cost of a candidate solution  $\mathcal{S}$  is denoted by  $C(\mathcal{S}, M) = \sum_{p \in \mathcal{P}} (l_p^P - e_p^P) + \sum_{n \in \mathcal{N}} (l_n^R - e_n^R) + M(\delta(\mathcal{S}) + \gamma(\mathcal{S}))$ , where  $M$  is a very large number. This implies that, for the purpose of detecting new best solutions, the total amount of infeasibilities is the first criterion, while the duration comes second if compared solutions are both feasible or if they have the same amount of infeasibilities.

However, when guiding the search, for the purpose of accepting or rejecting moves, an alternate cost function is used in order to ease exploring infeasible solutions:  $C(\mathcal{S}, \alpha) = \sum_{p \in \mathcal{P}} (l_p^P - e_p^P) + \sum_{n \in \mathcal{N}} (l_n^R - e_n^R) + \alpha(\delta(\mathcal{S}) + \gamma(\mathcal{S}))$ , where  $\alpha \in [\alpha_{min}, \alpha_{max}]$  is an adaptive parameter. The value of  $\alpha$  is initialized to  $\alpha_{min}$  and a parameter  $\mu \geq 1$  controls its variation as described in Olivera and Viera [2007]. Whenever an accepted solution contains overtime or time warp, the value of  $\alpha$  is set to  $\min\{\alpha\mu, \alpha_{max}\}$  to focus on reducing infeasibility. Else, the value of  $\alpha$  is set to  $\max\{\alpha/\mu, \alpha_{min}\}$  to encourage visiting infeasible solutions. After  $\xi$  iterations with  $\alpha$  set to  $\alpha_{max}$ , the value of  $\alpha$  is reset to  $\alpha_{min}$  to prevent it from remaining stuck at its maximum value  $\alpha_{max}$ . The values of  $\alpha_{min}$ ,  $\alpha_{max}$ ,  $\mu$ , and  $\xi$  are determined during the algorithm configuration phase.

### 4.3 Evaluation subproblem

In heuristic approaches, being able to evaluate moves efficiently is crucial to ensure the overall speed of the method. In this problem, however, performing a move that modifies the schedule of any nurse or pharmacist leads to a chain of reactions that impacts the complete solution schedule. This is due to the interdependence between the operations to be performed by pharmacists and nurses. Let us consider, for example, a move that inserts a patient inside a given trip. Such an insertion affects the administration start times during the nurse's journey. Consequently, the production start times of the drugs administered by the concerned nurse are impacted due to the linking constraints between production and routing: drug stability constraints and trip start time constraints. Thus, all the pharmacists producing these drugs see their respective production schedule affected. Similarly, each change in the production schedule impacts the administration schedule. Such interdependencies are usually encountered in vehicle routing problems with synchronization, surveyed in Drexel [2012], and tend to complicate the evaluation of candidate solutions.

Given a removal or insertion move to evaluate, the production and administration sequences of the candidate solution are known but the optimal values of all the start time variables need to be computed: the complete production and

administration schedules have to be determined. We call this issue the evaluation subproblem. As explained further in this section, this evaluation subproblem can be solved using a linear programming (LP) model.

Very few recent studies have tried to evaluate efficiently moves in problems that include interdependencies. In Masson et al. [2014], handicapped persons must be transported from their origin location to their destination through a dedicated fleet. These persons may have to be transferred from one vehicle to another at so-called transfer locations, implying precedence constraints, and hence interdependence, between vehicle journeys. The authors model the evaluation of the feasibility of a move as a simple temporal problem (STP), whose constraints impose bounds on time intervals separating pairs of events. In Masson et al. [2013], the concept of forward time slack of Savelsbergh [1992] is extended to evaluate efficiently the feasibility of moves for the same problem. The complexity of evaluating the feasibility of inserting a customer inside a vehicle journey is cubic in the number of customers in the case of Masson et al. [2014] and it is performed in constant time with quadratic updates when a solution is modified in Masson et al. [2013]. Lehuédé et al. [2015] extend the technique presented in Masson et al. [2013] for two more variants of vehicle routing with synchronization constraints.

However, in this work, those techniques are not straightforward to adapt for the following reasons.

- We are not only interested in evaluating the feasibility of the moves, but also the change in the objective function value. Indeed, in our case, changing the schedule of the solution also implies a change in the total working duration.
- Since feasible schedules may be difficult to obtain, and because we want to allow visiting, during the search, solutions that are infeasible w.r.t. time-related constraints, we need to evaluate the new working duration of a candidate solution as well as its amount of infeasibility.

Let us now present the evaluation subproblem considered in this work. Let  $\mathcal{X}$  be a set of production and drug administration sequences which include all the operations to be executed. Given this set  $\mathcal{X}$ , the evaluation subproblem is to determine the optimal start and completion times of the production, administration, and loading operations. The obtained schedule is denoted  $\mathcal{S}_{opt}(\mathcal{X})$ . We model this problem as a linear program. The sets, parameters, and decision variables of this model are provided in Table 1.

Even though there are no time windows directly imposed on the start times of the production operations in the studied problem, the earliest production start time of a drug  $i$  is clearly constrained due to its earliest administration start time and its stability time. Moreover, the latest production start time of a drug  $i$  has to permit starting the drug administration before the end of the corresponding

<b>Sets</b>	
$\mathcal{J}$ , indexed by $i$ and $j$	Set of patients
$\mathcal{J}_p$	Set of production operations executed by pharmacist $p \in \mathcal{P}$
$\mathcal{J}_n$	Set of patients visited by nurse $n \in \mathcal{N}$
$\mathcal{R}$ , indexed by $r$ and $s$	Set of non empty trips
$\mathcal{A}_P = \{(i, j) \in \mathcal{J}^2 \mid i \text{ is produced right before } j\}$	Set of consecutive production operations executed by a same pharmacist
$\mathcal{A}_R = \{(i, j) \in \mathcal{J}^2 \mid i \text{ is administered right before } j\}$	Set of consecutive patients visited by a same nurse in a same trip
$\mathcal{A}_N = \{(r, s) \in \mathcal{R}^2 \mid r \text{ is executed right before } s\}$	Set of consecutive trips performed by a same nurse
<b>Parameters</b>	
$P_i$	Processing time of drug $i \in \mathcal{J}$
$ST_i$	Stability time of drug $i \in \mathcal{J}$
$S_i$	Time necessary to administrate drug $i \in \mathcal{J}$
$E_i$	Earliest start time of administration operation $i \in \mathcal{J}$
$L_i$	Latest start time of administration operation $i \in \mathcal{J}$
$E_i^P$	Earliest start time of production operation $i \in \mathcal{J}$
$L_i^P$	Latest start time of production operation $i \in \mathcal{J}$
$T_{ij}$	Travel time between patient $i$ and $j \in \mathcal{J}$
$T_{0i}$ (resp. $T_{i0}$ )	Travel time between the hospital and patient $i \in \mathcal{J}$ (resp. patient $i$ and the hospital)
$D^{max}$	Maximum length of a shift
$first_p^P$	First production operation executed by pharmacist $p \in \mathcal{P}$
$last_p^P$	Last production operation executed by pharmacist $p \in \mathcal{P}$
$first_n^T$	First trip performed by nurse $n \in \mathcal{N}$
$last_n^T$	Last trip performed by nurse $n \in \mathcal{N}$
$first_r^R$	First patient visited in trip $r \in \mathcal{R}$
$last_r^R$	Last patient visited in trip $r \in \mathcal{R}$
$trip_i$	Trip that visits patient $i \in \mathcal{J}$
$M$	A very large number
<b>Decision variables</b>	
$t_i^P$	Start time of production operation $i \in \mathcal{J}$
$t_i^R$	Start time of administration operation $i \in \mathcal{J}$
$e_p^P$	Start time of the working shift of pharmacist $p \in \mathcal{P}$
$l_p^P$	End time of the working shift of pharmacist $p \in \mathcal{P}$
$e_r^R$	Start time of trip $r \in \mathcal{R}$
$l_r^R$	End time of trip $r \in \mathcal{R}$
$\delta_i^P$	Time warp on the start time of production operation $i \in \mathcal{J}$
$\delta_i^R$	Time warp on the start time of administration operation $i \in \mathcal{J}$
$\gamma_p^P$	Overtime of pharmacist $p \in \mathcal{P}$
$\gamma_n^R$	Overtime of nurse $n \in \mathcal{N}$

Table 1: Notations for the LP formulation of the evaluation subproblem.

time window. Taking those dependencies into account, we deduce a time window  $[E_i^P, L_i^P]$  that is valid on the start time of each production operation  $i \in \mathcal{J}$ :

$$\begin{aligned} E_i^P &= E_i - ST_i, \\ L_i^P &= L_i - T_{0i} - P_i. \end{aligned}$$

With the aim of determining the amount of infeasibility related to these bounds, we impose time window constraints on the production start times in the models provided here after. Below, we first give the LP formulation of the evaluation subproblem if infeasibilities are not allowed.

$$\min \sum_{p \in \mathcal{P}} (l_p^P - e_p^P) + \sum_{n \in \mathcal{N}} (l_{last_n^T}^R - e_{first_n^T}^R) \quad (1)$$

$$s.t. \quad t_i^P + P_i \leq t_j^P \quad \forall (i, j) \in \mathcal{A}_P \quad (2)$$

$$t_i^R + S_i + T_{ij} \leq t_j^R \quad \forall (i, j) \in \mathcal{A}_R \quad (3)$$

$$E_i^P \leq t_i^P \leq L_i^P \quad \forall i \in \mathcal{J} \quad (4)$$

$$E_i \leq t_i^R \leq L_i \quad \forall i \in \mathcal{J} \quad (5)$$

$$t_i^R - t_i^P \leq ST_i \quad \forall i \in \mathcal{J} \quad (6)$$

$$t_i^P + P_i \leq e_{trip_i}^R \quad \forall i \in \mathcal{J} \quad (7)$$

$$e_p^P \leq t_{first_p^P}^P \quad \forall p \in \mathcal{P} \quad (8)$$

$$t_{last_p^P}^P + P_{last_p^P} \leq l_p^P \quad \forall p \in \mathcal{P} \quad (9)$$

$$e_r^R + T_{0,first_r^R} \leq t_{first_r^R}^R \quad \forall r \in \mathcal{R} \quad (10)$$

$$t_{last_r^R}^R + S_{last_r^R} + T_{last_r^R,0} \leq l_r^R \quad \forall r \in \mathcal{R} \quad (11)$$

$$l_r^R \leq e_s^R \quad \forall (r, s) \in \mathcal{A}_N \quad (12)$$

$$l_p^P - e_p^P \leq D^{max} \quad \forall p \in \mathcal{P} \quad (13)$$

$$l_{last_n^T}^R - e_{first_n^T}^R \leq D^{max} \quad \forall n \in \mathcal{N} \quad (14)$$

$$t_i^P, t_i^R \geq 0 \quad \forall i \in \mathcal{J} \quad (15)$$

$$e_p^P, l_p^P \geq 0 \quad \forall p \in \mathcal{P} \quad (16)$$

$$e_r^R, l_r^R \geq 0 \quad \forall r \in \mathcal{R} \quad (17)$$

The objective function to be minimized (1) is the total working time of pharmacists and nurses. Constraints (2) and (3) respectively ensure the consistency of production and administration start times. Constraints (4) and (5) are time windows constraints. Constraints (6) and (7) are the synchronization constraints that link production and administration operations: the stability of each drug

must be respected and each trip may start only if all the concerned drugs are produced. Constraints (8) and (9) respectively determine the start and end time of the working shift of each pharmacist. Similarly, Constraints (10) and (11) respectively determine the start and end time of each trip. Constraints (12) state that a trip may not start before the end of the preceding trip performed by the same nurse. Constraints (13) and (14) impose a maximum working duration for the pharmacists and nurses. Finally, Constraints (15) to (17) define the variable domains.

In order to evaluate the amount of infeasibility of a given solution, we first define non-negative time warp variables for the start times of production and administration operations. The time warp  $\delta_i^R$  (resp.  $\delta_i^P$ ) of the administration start time (resp. production start time) of drug  $i$  is the quantity of time units that the concerned nurse (resp. pharmacist) would need to travel back in time in order to start the operation just on time. That is, the fictive administration start time,  $f_i^R = t_i^R - \delta_i^R$ , and the fictive production start time,  $f_i^P = t_i^P - \delta_i^P$ , of a drug  $i \in \mathcal{J}$  always satisfy Constraints (4)-(7). Every pharmacist and nurse continue their sequence according to the fictive start times of the already executed operations. In order to integrate the time warp variables into the model, Constraints (2)-(11) are replaced by the following:

$$t_i^P - \delta_i^P + P_i \leq t_j^P \quad \forall (i, j) \in \mathcal{A}_P \quad (18)$$

$$t_i^R - \delta_i^R + S_i + T_{ij} \leq t_j^R \quad \forall (i, j) \in \mathcal{A}_R \quad (19)$$

$$E_i^P \leq t_i^P - \delta_i^P \leq L_i^P \quad \forall i \in \mathcal{J} \quad (20)$$

$$E_i \leq t_i^R - \delta_i^R \leq L_i \quad \forall i \in \mathcal{J} \quad (21)$$

$$t_i^R - \delta_i^R - ST_i \leq t_i^P \quad \forall i \in \mathcal{J} \quad (22)$$

$$t_i^P - \delta_i^P + P_i \leq e_{trip_i}^R \quad \forall i \in \mathcal{J} \quad (23)$$

$$t_{last_p^P}^P - \delta_{last_p^P}^P + P_{last_p^P} \leq l_p^P \quad \forall p \in \mathcal{P} \quad (24)$$

$$t_{last_r^R}^R - \delta_{last_r^R}^R + S_{last_r^R} + T_{last_r^R,0} \leq l_r^R \quad \forall r \in \mathcal{R} \quad (25)$$

In order to allow each nurse or pharmacist to fictively work more than the maximum authorized duration  $D^{max}$ , non-negative overtime variables are introduced into the model. The overtime of each pharmacist  $p \in \mathcal{P}$  (resp. each nurse  $n \in \mathcal{N}$ ) is represented by  $\gamma_p^P$  (resp.  $\gamma_n^R$ ). The overtime variables are added in Constraints (13) and (14) as follows:

$$l_p^P - e_p^P + \sum_{i \in \mathcal{J}_p} \delta_i^P \leq D^{max} + \gamma_p^P \quad \forall p \in \mathcal{P} \quad (26)$$

$$l_{last_n^R}^R - e_{first_n^R}^R + \sum_{i \in \mathcal{J}_n} \delta_i^R \leq D^{max} + \gamma_n^R \quad \forall n \in \mathcal{N} \quad (27)$$

The aim here is to obtain the production and routing schedules that have the least amount of infeasibility. The objective function of the model is modified accordingly, i.e., the total amount of infeasibility is penalized by  $M$ , a very large number, in (1):

$$\begin{aligned} \min \quad & \sum_{p \in \mathcal{P}} (l_p^P - e_p^P) + \sum_{n \in \mathcal{N}} (l_{last_n^R}^R - e_{first_n^R}^R) \\ & + M \sum_{i \in \mathcal{J}} (\delta_i^P + \delta_i^R) + M \left( \sum_{p \in \mathcal{P}} \gamma_p^P + \sum_{n \in \mathcal{N}} \gamma_n^R \right) \end{aligned} \quad (28)$$

In our LNS algorithm, the LP model that allows infeasible schedules is solved whenever the optimal schedule  $\mathcal{S}_{opt}(\mathcal{X})$  of a given set  $\mathcal{X}$  of production and administration sequences needs to be computed.

#### 4.4 Heuristic evaluation

In the evaluation subproblem presented above, the complete schedule of a candidate solution needs to be computed to determine the exact objective function value of this solution. However, this value may be approximated by fixing some parts of the schedule in order to avoid incessant resolutions of the above LP. If the production schedule is considered as fixed, i.e., the start times of all the production operations are fixed, then moves on the administration schedule may be evaluated in constant time using the concatenation equations described in Vidal et al. [2015], only slightly adapted as explained later in this section. Likewise, if the administration schedule is considered as fixed, i.e., the start times of the trips and the drug administration operations are fixed, then moves on the production schedule may be evaluated in constant time. Of course, as already mentioned, fixing a part of the schedule leads to obtaining a suboptimal objective function value for the given production and administration sequences.

**Dynamic time windows.** Given a production schedule  $\mathcal{S}^P$ , the fictive production start time  $\hat{f}_i^P$  is known for all  $i \in \mathcal{J}$ , implying time bounds on the drug administration operations.

- The administration start time of drug  $i$  is bounded from above because of the stability time:  $t_i^R \leq \min \{ \hat{f}_i^P + ST_i; L_i \}$ .
- The administration start time of drug  $i$  is bounded from below because of the processing and traveling times:  $t_i^R \geq \max \{ \hat{f}_i^P + P_i + T_{0i}; E_i \}$ .



Thus, whenever a production schedule is given, the administration time windows  $[E_i, L_i]$  may be strengthened. We refer to these induced time windows as *dynamic* time windows. The dynamic time window  $[\hat{E}_i^R, \hat{L}_i^R]$  of an administration operation  $i \in \mathcal{J}$  is calculated as follows:

$$\begin{aligned}\hat{E}_i^R &= \max \{ \hat{f}_i^P + P_i + T_{0i}; E_i \}, \\ \hat{L}_i^R &= \min \{ \hat{f}_i^P + ST_i; L_i \}.\end{aligned}$$

Similarly, given a routing schedule  $\mathcal{S}^R$ , the administration start time  $\hat{f}_i^R$  of drug  $i$  and the start time  $\hat{e}_{trip_i}^R$  of the trip visiting patient  $i$  are known for all  $i \in \mathcal{J}$ , implying time bounds on the production operations.

- The production start time of drug  $i$  is bounded from below because of the stability time:  $t_i^P \geq \hat{f}_i^R - ST_i$ .
- The production start time of drug  $i$  is bounded from above by the start time of the trip that visits patient  $i$ :  $t_i^P \leq \hat{e}_{trip_i}^R - P_i$ .

Consequently, the dynamic time window  $[\hat{E}_i^P, \hat{L}_i^P]$  of a production operation  $i \in \mathcal{J}$  is determined as follows:

$$\begin{aligned}\hat{E}_i^P &= \hat{f}_i^R - ST_i, \\ \hat{L}_i^P &= \hat{e}_{trip_i}^R - P_i.\end{aligned}$$

**Concatenation equations.** We use the concatenation equations proposed by Vidal et al. [2015] in order to evaluate moves of one subproblem given time windows - dynamic or not - imposed on its operations. Vidal et al. [2015] describe each move as a series of path concatenations, which allows evaluating vertex-based or edge-based moves in  $\mathcal{O}(1)$ , even though infeasible solutions are considered. A path is defined as a sequence of vertices, where each vertex represents a single operation. An implicit timing of a path  $\tau$  is described through four measures: the minimum duration  $D(\tau)$ , the minimum time warp usage  $TW(\tau)$ , the earliest start time  $E(\tau)$  of the first operation  $\tau(1)$ , and the latest start time  $L(\tau)$  of the first operation  $\tau(1)$ : for  $\tau(1)$ , any start time in the interval  $[E(\tau), L(\tau)]$  results in obtaining the minimum duration and the minimum time warp usage for path  $\tau$ .

Let  $\tau_1$  and  $\tau_2$  be two paths. Define  $\Lambda_{ij}$  as the minimum delay between the completion of operation  $i \in \mathcal{J}$  and the start of operation  $j \in \mathcal{J}$ . The four

measures for  $\tau_1 \oplus \tau_2$ , i.e., the concatenation of  $\tau_1$  and  $\tau_2$ , can be computed using the concatenation equations of Vidal et al. [2015] reported below.

$$\Delta = D(\tau_1) - TW(\tau_1) + \Lambda_{\tau_1(|\tau_1|), \tau_2(1)} \quad (29)$$

$$\Delta WT = \max\{E(\tau_2) - \Delta - L(\tau_1), 0\} \quad (30)$$

$$\Delta TW = \max\{E(\tau_1) + \Delta - L(\tau_2), 0\} \quad (31)$$

$$D(\tau_1 \oplus \tau_2) = D(\tau_1) + D(\tau_2) + \Lambda_{\tau_1(|\tau_1|), \tau_2(1)} + \Delta WT \quad (32)$$

$$E(\tau_1 \oplus \tau_2) = \max\{E(\tau_2) - \Delta, E(\tau_1)\} - \Delta WT \quad (33)$$

$$L(\tau_1 \oplus \tau_2) = \min\{L(\tau_2) - \Delta, L(\tau_1)\} + \Delta TW \quad (34)$$

$$TW(\tau_1 \oplus \tau_2) = TW(\tau_1) + TW(\tau_2) + \Delta TW \quad (35)$$

**Concatenations for production moves.** When considering the production subproblem, a path represents a sequence of production operations performed by the same pharmacist. For a path  $\tau$  containing the production operation of a single drug  $i$ ,  $D(\tau)$  and  $TW(\tau)$  are respectively equal to  $P_i$  and 0. Depending if the production time window of  $i$  considered during the concatenation operation is dynamic or not,  $E(\tau)$  and  $L(\tau)$  are respectively equal either to  $\hat{E}_i^P$  and  $\hat{L}_i^P$ , or to  $E_i^P$  and  $L_i^P$ . Note that  $\Lambda_{ij} = 0$  for all  $i \in \mathcal{J}$  and  $j \in \mathcal{J} \setminus \{i\}$ .

**Concatenations for routing moves.** When considering the routing subproblem, a path represents a sequence of administration and loading operations performed by the same nurse. In this study, concatenations of administration sequences are always performed given a production schedule  $\mathcal{S}^P$ . Consequently, only dynamic time windows are used. The vertex set  $\bar{\mathcal{J}} = \mathcal{J} \cup \{0\}$  represents all the administration operations to be executed together with the loading operations at the hospital. For all  $i \in \bar{\mathcal{J}}$  and  $j \in \bar{\mathcal{J}} \setminus \{i\}$ , the minimum delays are defined as  $\Lambda_{ij} = T_{ij}$ . For a path  $\tau$  containing a single patient  $i$ , the four measures  $D(\tau)$ ,  $TW(\tau)$ ,  $E(\tau)$ , and  $L(\tau)$  are respectively equal to  $S_i$ , 0,  $\hat{E}_i^R$ , and  $\hat{L}_i^R$ . However, when concatenating a path  $\tau_1$  that only contains a loading operation with a path  $\tau_2$  starting with one or more patient visits, we have to consider those patients to determine the earliest start time of path  $\tau_1$ . The start time of a trip  $r$  is bounded from below by the production completion times of the drugs administered during the trip:  $e_r^R \geq \max_{i \in \mathcal{J}_r} \{\hat{f}_i^P + P_i\}$ , where  $\mathcal{J}_r \subseteq \mathcal{J}$  is the subset of drugs administered in trip  $r$ . Thus, the earliest start time  $\hat{E}_r^T$  of a trip  $r$  depends on the subset of patients visited and is defined as follows:

$$\hat{E}_r^T = \max_{i \in \mathcal{J}_r} \{\hat{f}_i^P + P_i\}$$

Consequently, the four measures  $D(\tau)$ ,  $TW(\tau)$ ,  $E(\tau)$ , and  $L(\tau)$  of a path  $\tau$  containing only a loading operation at the hospital, which is followed by trip  $r$  in the considered concatenation, are respectively 0, 0,  $\hat{E}_r^T$ , and  $L_0$ .

In order to maintain the values needed to evaluate moves in  $\mathcal{O}(1)$  when using multi-trip operators with trip-dependent earliest loading start times, we employ the path concatenations proposed by François et al. [2019]. Let  $\mathcal{T}$  be a journey consisting of one or more trips. Let  $o$  be the first loading operation of  $\mathcal{T}$  and  $o'$  be its final visit at the hospital. For each vertex  $i$  in  $\mathcal{T}$ , let  $O_i$  and  $O'_i$  denote respectively the visits at the hospital which initiate and terminate the trip containing  $i$ . Note that  $o$  is equivalent to  $O_i$  if vertex  $i$  belongs to the first trip of  $\mathcal{T}$ . Similarly,  $o'$  is equivalent to  $O'_i$  if  $i$  belongs to the last trip of  $\mathcal{T}$ . When updating  $\mathcal{T}$ , for each  $i$  in  $\mathcal{T}$ , we maintain the previously defined four measures for four different paths:

- $\tau_{[o,i]}$ , from the first loading operation of  $\mathcal{T}$  included to vertex  $i$  included,
- $\tau_{[i,o']}$ , from vertex  $i$  included to the last loading operation of  $\mathcal{T}$  included,
- $\tau_{(O_i,i]}$ , from the last visit at the hospital preceding vertex  $i$  excluded to vertex  $i$  included,
- $\tau_{[i,O'_i)}$ , from vertex  $i$  included to the first visit at the hospital following vertex  $i$  excluded.

The earliest start time  $\hat{E}_r^T$  of trip  $r$  that starts at  $O_i$  and ends at  $i$  is also maintained. The first three path types are used to evaluate insertion and removal moves. The fourth one is needed for merging consecutive trips. When inserting or removing an administration operation, the earliest start time of the preceding visit at the hospital changes. Maintaining the four measures of those four different paths for each  $i$  in  $\mathcal{T}$  permits evaluating any insertion or removal move in  $\mathcal{O}(1)$ . We refer the reader to François et al. [2019] for an in-depth explanation regarding move executions.

In the following, we denote by  $C^R(\mathcal{X}^R, M)$  or  $C^R(\mathcal{X}^R, \alpha)$  the minimum cost of a set of administration sequences  $\mathcal{X}^R$  obtained by summing the minimum possible duration of each sequence, and penalizing with a factor  $M$  or  $\alpha$  the total associated overtime and time warp.

## 4.5 Structure of the LNS algorithm

The proposed LNS is a destroy & repair heuristic composed of two embedded loops. The outer loop perturbs the incumbent solution by modifying its production sequences. Then, the production sequences are fixed while the inner loop concentrates on improving the administration sequences. Within the inner loop, a

production schedule is known at all times and dynamic administration time windows are imposed. Below, the creation of the initial solution is described, as well as the details of the two embedded loops. Algorithm 1 then presents the outline of the proposed LNS algorithm.

**Initial solution.** The initial solution is created in three phases: first, the production schedule is initialized, second, the administration sequences are created, and third, the optimal schedule of production and administration sequences is determined using the LP procedure described in Section 4.3.

To create the production schedule, we proceed in two steps: an assignment step, where drugs are assigned to pharmacists, and a scheduling step, where production tasks of each pharmacist are scheduled.

In the assignment step, the drugs to produce are first sorted in increasing order of their *tightness* defined as  $ST_i - P_i - T_{0i}$ . The tightness represents the maximum slack of the administration start time with respect to the drug production start time. The sorted list of drugs is split into  $|\mathcal{P}|$  *segments*, each of size  $\lfloor (|\mathcal{J}|/|\mathcal{P}|) \rfloor$  or  $\lceil (|\mathcal{J}|/|\mathcal{P}|) \rceil$ . The drugs in each segment are then sorted by decreasing processing times, and segments are concatenated in a single list starting from the first segment up to the last one. Finally, drug in position  $i = 0, \dots, |\mathcal{J}| - 1$  of the list is assigned to the pharmacist  $i \bmod |\mathcal{P}|$ .

This first step is designed to create an even assignment of the drugs to the pharmacists with respect to the tightness, while taking into account at the same time the balance of the processing times.

After performing the assignment, the scheduling step begins. The drugs to be processed by each pharmacist are sorted in increasing order of the center of their static production time window, i.e.,  $(E_i^P + L_i^P)/2$ . This produces a production sequence for each pharmacist. The production of each drug is then scheduled at the latest possible start time that minimizes the working duration of each pharmacist while satisfying the static production time windows. Those computations are straightforward using the concatenation equations described in Section 4.4.

The dynamic time windows of the administration operations are set according to those schedules. Then, a vehicle-based regret insertion heuristic, as defined in François et al. [2019], creates the administration sequences. The LP procedure described in Section 4.3 is then used to determine the optimal schedule of the initial production and administration sequences.

**Outer loop.** While many different sets of production sequences may yield the same minimum production cost, routing costs are highly dependent on the administration sequences. This is the main reason why the algorithm perturbs the production sequences from time to time before evaluating many routing moves

given a fixed production sequence. In the following, we call *perturbation* the process of creating new schedules as a basis for the inner loop iterations.

In each outer loop iteration, one out of four perturbation schemes is used, with a dynamic parameter  $k$ . These four schemes all follow the same steps.

Step 1: Remove  $k$  production operations.

Step 2: Remove either the corresponding  $k$  administration operations, or all of them (depending on the perturbation scheme used).

Step 3: Reinsert the removed production operations.

Step 4: Compute a temporary production schedule and deduce the dynamic administration time windows.

Step 5: Reinsert the removed administration operations.

Step 6: Compute the optimal (production and administration) schedule of the new set of sequences.

In Step 1, those drugs, whose dynamic administration time window width is short compared to the width of their original time window, are more likely to be removed. The removal criteria are mathematically described in Section 4.6. In Step 2, the administration sequences are destroyed. Depending on the perturbation scheme, either the  $k$  drugs removed from the production sequences are also removed from the administration sequences (perturbation schemes 1 and 2), or all administration sequences are completely emptied (perturbation schemes 3 and 4).

For the purpose of reinserting the removed drug production operations into the production sequences during Step 3, production time windows should be considered. In perturbation schemes 1 and 2, since only the  $k$  drugs removed from the production sequences are also removed from the administration sequences, the dynamic production time windows of the removed drugs are reset to their original production time window value  $[E_i^P, L_i^P]$ , while the dynamic production time windows  $[\hat{E}_i^P, \hat{L}_i^P]$  of the other drugs are kept unchanged. This ensures that the new production sequences will be compatible with the partial administration sequences that were not destroyed. On the contrary, in schemes 3 and 4, since all the administration sequences are reinitialized, all the dynamic production time windows are reset to their original production time window values  $[E_i^P, L_i^P]$ . The reinsertion of the  $k$  removed drugs in the production sequences is performed with a greedy heuristic, which aims at widening the dynamic administration time windows compared to those of the last incumbent solution. This insertion heuristic is detailed in Section 4.6.

In order to reinsert the removed administration operations consistently with the production sequences, a temporary production schedule is computed in Step 4. Indeed, this allows the calculation of dynamic administration time windows that will favour coherence between the production and the administration sequences. Two options are considered. In schemes 1 and 3, we select the earliest possible start time within the start time interval that minimizes the makespan of the sequence. In schemes 2 and 4, the latest start time in the same interval is chosen. Once the production schedule is fixed, dynamic administration time windows  $[\hat{E}_i^R, \hat{L}_i^R]$  are updated for all drugs. In Step 5, insertion of unadministered drugs (either  $k$  or all drugs) is performed using a greedy routing heuristic.

Finally, in Step 6, the optimal schedule of the new set of sequences is computed thanks to the dedicated LP procedure. The new perturbed solution created through this process is the initial solution of the inner loop described below.

At the beginning of the algorithm,  $k$  and the perturbation scheme are equal to 1. The perturbation schemes are used in a cyclic fashion (1, 2, 3, 4, 1, etc.). After the fourth perturbation scheme has been used, the value of  $k$  is increased by one unit or it is set to 1 if the new value would exceed  $k_{max}$ . Whenever a new best solution is detected, it becomes the new incumbent of the outer loop, and the value of  $k$  is reset to 1 as well as the perturbation scheme. The value of  $k_{max}$  is determined during the configuration phase as a fraction of the instance size:  $k_{max} = \lfloor \lambda_{outer} \times |\mathcal{J}| \rfloor$ , where  $\lambda_{outer} \in ]0, 1]$ .

The stopping criterion of the outer loop is the maximum run time of the algorithm.

**Inner loop.** During an outer loop iteration, after a perturbation has been applied to the incumbent solution using the six steps detailed above, the inner loop iteratively performs constrained routing moves, i.e., moves that modify only the administration sequences while keeping the production schedule unchanged. That is, a production schedule  $\mathcal{S}^P$  is provided at all times, and the dynamic time windows imposed by  $\mathcal{S}^P$  on the administration start times are those that will be taken into account to evaluate the routing moves. As stated in Section 4.2, infeasible routing moves may be considered for acceptance. The feasibility of administration sequences is evaluated relatively to the dynamic administration time windows.

At each iteration of the inner loop,  $q$  drug administration operations are removed and reinserted using randomly selected removal and insertion heuristics described in Section 4.6. Note that all removal (resp. insertion) heuristics have the same probability to be chosen. Indeed, as shown in François et al. [2016] and Turkeš et al. [2021], adaptive heuristic selection is not necessarily an important algorithmic component, especially if offline parameter configuration is used, which is the case here. In the first iteration, and each time a solution is accepted, the

value of  $q$  is set to 1. If the candidate solution is rejected, the value of  $q$  is increased by one unit, as long as it does not exceed  $q_{max}$ , in which case it is set to  $q_{low}$ . The values of  $q_{max}$  and  $q_{low}$  are determined during the configuration phase as functions of the instance size:  $q_{max} = \lfloor \lambda_{inner} \times |\mathcal{J}| \rfloor$  and  $q_{low} = \lfloor \lambda_{inner} \times |\mathcal{J}| / \delta \rfloor$ , where  $\lambda_{inner} \in ]0, 1]$  and  $\delta \in \mathbb{N}_0^+$ .

In a given inner loop iteration, the selected heuristics modify  $\mathcal{X}_{inner}^R$ , the set of administration sequences of  $\mathcal{S}_{inner}$ , to obtain a candidate set of administration sequences  $\mathcal{X}'_{inner}^R$ . A simulated annealing framework is used to decide whether to accept  $\mathcal{X}'_{inner}^R$ . The acceptance probability is equal to  $\exp[-(C^R(\mathcal{X}'_{inner}^R, \alpha) - C^R(\mathcal{X}_{inner}^R, \alpha)) / (\theta C^R(\mathcal{X}_{inner}^R, \alpha))]$ , where  $\theta$  is the temperature. If the candidate set of administration sequences is accepted, then  $\mathcal{S}_{opt}(\mathcal{X}'_{inner}^R)$  is computed thanks to the LP procedure, and it becomes the inner loop incumbent. At each iteration,  $\theta$  is set to  $\max\{\eta\theta, \theta_{min}\}$ , where  $\eta \in [0, 1]$  is the cooling factor. The minimum temperature is defined as  $\theta_{min} = \kappa\theta_0$  with  $\kappa \in [0, 1]$ . The configuration phase determines the respective value of  $\theta_0$ ,  $\kappa$ , and  $\eta$ .

In order to detect new best solutions, the cost measure of a solution  $\mathcal{S}$  becomes  $C(\mathcal{S}, M)$ . Indeed, the value returned by  $C(\mathcal{S}, \alpha)$  does not depend only on the characteristics of the solution  $\mathcal{S}$  but also on the value of the adaptive parameter  $\alpha$  during a given iteration. On the contrary, the value returned by  $C(\mathcal{S}, M)$  stays consistent throughout the course of the algorithm. Thus, once a solution has been accepted in the inner loop and recorded as the new  $\mathcal{S}_{inner}$ ,  $C(\mathcal{S}_{inner}, M)$  is compared to  $C(\mathcal{S}_{outer}, M)$ , which is always the best cost found so far.

The stopping criterion of the inner loop is met if the maximum run time of the algorithm is reached, if the consecutive number of infeasible solutions exceeds  $\omega$ , or if the consecutive number of non improving solutions exceeds  $\omega'$ . The respective values of  $\omega$  and  $\omega'$  are determined during the configuration phase.

## 4.6 Removal and insertion heuristics

Several types of heuristic moves are used in the above described algorithm.

- Removal of production operations (outer loop).
- Insertion of production operations (outer loop).
- Removal of administration operations (inner loop).
- Insertion of administration operations (inner loop).

**Removal heuristic for production sequences.** The first step of the outer loop perturbation is to remove  $k$  production operations from the production sequences. For each patient, the value  $(\hat{L}_i^R - \hat{E}_i^R) / (L_i - E_i)$  is computed. It measures

---

**Algorithm 1** Large neighborhood search heuristic.

---

- 1: Construct an initial set of production and administration sequences  $\mathcal{X}_{init}$
  - 2: Compute the optimal schedule  $\mathcal{S}_{init} \leftarrow \mathcal{S}_{opt}(\mathcal{X}_{init})$
  - 3: Update  $\hat{E}_i^P$ ,  $\hat{L}_i^P$ ,  $\hat{E}_i^R$ , and  $\hat{L}_i^R$  for all  $i \in \mathcal{J}$
  - 4: Set  $k \leftarrow 1$ ; set  $\mathcal{S}_{outer} \leftarrow \mathcal{S}_{init}$
  - 5: **while** the stopping criterion is not met (outer loop) **do**
  - 6:   Apply perturbation to  $\mathcal{S}_{outer}$  (Steps 1 to 6) to obtain  $\mathcal{S}_{inner}$
  - 7:   Update  $k$  and perturbation scheme
  - 8:   **if**  $C(\mathcal{S}_{inner}, M) < C(\mathcal{S}_{outer}, M)$  **then**
  - 9:     Set  $\mathcal{S}_{outer} \leftarrow \mathcal{S}_{inner}$ ; reset  $k$  and perturbation scheme
  - 10:   **end if**
  - 11:   Set  $q \leftarrow 1$
  - 12:   **while** the stopping criterion is not met (inner loop) **do**
  - 13:     Select  $h_{rem}$  and  $h_{ins}$  randomly
  - 14:     Obtain  $\mathcal{X}'_{inner}$  by using  $h_{rem}$  and  $h_{ins}$  on  $\mathcal{X}_{inner}^R$
  - 15:     **if**  $C(\mathcal{X}'_{inner}, \alpha)$  satisfies the SA acceptance criteria **then**
  - 16:       Set  $\mathcal{S}_{inner} \leftarrow \mathcal{S}_{opt}(\mathcal{X}'_{inner})$
  - 17:       Update  $\hat{E}_i^P$ ,  $\hat{L}_i^P$ ,  $\hat{E}_i^R$ , and  $\hat{L}_i^R$  for all  $i \in \mathcal{J}$
  - 18:       Reset  $q$ ; update  $\alpha$
  - 19:       **if**  $C(\mathcal{S}_{inner}, M) < C(\mathcal{S}_{outer}, M)$  **then**
  - 20:         Set  $\mathcal{S}_{outer} \leftarrow \mathcal{S}_{inner}$ ; reset  $k$  and perturbation scheme
  - 21:       **end if**
  - 22:     **else**
  - 23:       Update  $q$
  - 24:     **end if**
  - 25:   **end while**
  - 26: **end while**
-



the narrowing of the administration time window imposed by the start time of the production operation. Patients are sorted in a ranked list in increasing order of those values. The randomized selection mechanism proposed in Ropke and Pisinger [2006] is applied such that patients with small ranks have a higher probability to be removed (see Algorithm 2). The value of  $v^P$  is set during the configuration phase. If it is equal to 1, all the patients have the same probability to be selected. The probability of selecting a patient at the beginning of the ranked list increases when  $v^P$  is large.

---

**Algorithm 2** Randomized selection

---

- 1: Create a ranked list  $\mathcal{V}$  containing all the patients
  - 2: **while** the number of selected patients is strictly smaller than  $k$  **do**
  - 3:   Set the value of  $\rho \sim \mathcal{U}[0, 1[$
  - 4:   Select the patient whose index in  $\mathcal{V}$  is equal to  $\lfloor \rho^{v^P} \times |\mathcal{V}| \rfloor$ , where  $v^P \geq 1$  is a parameter.
  - 5:   Remove the selected patient from  $\mathcal{V}$
  - 6: **end while**
- 

**Insertion heuristic for production sequences.** In the outer loop, an insertion heuristic reconstructs the production sequences after they have been destroyed. Drugs are inserted one by one following a greedy mechanism. As already mentioned, many different sets of production sequences may yield the same cost. This is why the greedy criterion takes into account the potential impact of the insertion of a production operation on the administration sequences, rather than simply considering the duration increase of the production sequences.

The measure  $C^P(\mathcal{X}_p^P)$  related to the production sequence  $\mathcal{X}_p^P$  of pharmacist  $p$  is used as an indicator of the size of the dynamic routing time windows that result from  $\mathcal{X}_p^P$ . Let  $C^P(\mathcal{X}_p^P)$  be defined as  $\sum_{i \in \mathcal{X}_p^P} ((\bar{l}_i^R - \bar{e}_i^R) \times (l_i^R - \underline{e}_i^R)) / (L_i - E_i)$ , where  $\bar{l}_i^R - \bar{e}_i^R$  (resp.  $l_i^R - \underline{e}_i^R$ ) is the size of the dynamic administration time window when production start times are fixed at their latest (resp. earliest) possible values which minimize the duration of  $\mathcal{X}_p^P$ . The larger the size of the dynamic routing time windows, the higher the value of  $C^P(\mathcal{X}_p^P)$ . Let  $C^P(\mathcal{X}^P)$  be equal to  $\sum_{p \in \mathcal{P}} C^P(\mathcal{X}_p^P)$ , and let  $\Delta C_j^P(\mathcal{X}^P)$  be the variation of  $C^P(\mathcal{X}^P)$  if drug  $j$  is inserted in  $\mathcal{X}^P$  at the position resulting in the highest  $C^P(\mathcal{X}^P)$  value after insertion. The drug to select at each iteration of the greedy production insertion heuristic is the one that globally decreases the less the size of the time windows, i.e.,  $\operatorname{argmax}_{j \in \mathcal{J}^P} \{\Delta C_j^P(\mathcal{X}^P)\}$ , where  $\mathcal{J}^P$  denotes the set of drugs that need to be inserted in production sequences. Indeed,  $\Delta C_j^P(\mathcal{X}^P)$  is negative for all  $j$ , since inserting a customer in a production

sequence may never result in increasing the quantities  $(\bar{l}_i^R - \bar{e}_i^R)$  and  $(l_i^R - e_i^R)$ . Thus,  $\Delta C_j^P(\mathcal{X}^P)$  should be maximized.

**Removal and insertion heuristics for administration sequences.** Administration moves modify only the administration sequences without modifying the production schedule. That is, a given production schedule  $\mathcal{S}^P$  is considered as fixed and administration operations are removed and reinserted taking into account the dynamic time windows imposed by  $\mathcal{S}^P$  on the administration start times.

As stated before, the multi-trip operators proposed in François et al. [2016] are employed for that purpose. When inserting an administration task in a nurse’s journey, we consider the following four multi-trip insertion schemes and choose the least costly:

1. Insert the administration task.
2. Insert a reloading operation at the hospital and then the administration task.
3. Insert the administration task and then a reloading operation at the hospital.
4. Insert a new trip, which contains only the considered administration task between two reloading operations.

In removal heuristics, when removing an administration task would cause a journey to contain two consecutive reloading operations, one of these is also removed. A merge operator, which is included by default within all removal heuristics, allows merging two consecutive trips of a journey by removing the reloading operation between them. Merges are recursively performed if needed. The insertion and removal moves are evaluated in  $\mathcal{O}(1)$  using path concatenations as explained in Section 4.4. The heuristics selected in the inner loop are those that have proven useful in the ALNS algorithm of François et al. [2019] when the considered objective is the minimization of the working duration, i.e., random removal, worst removal - versions (a) and (b), Shaw removal - versions (b) and (c), greedy insertion, vehicle-based regret insertion, and position-based regret insertion.

As it was the case for the removal heuristic dedicated to production sequences, a randomization factor, named  $v^R$  is used in several removal heuristics cited above. Its value is determined during the configuration phase.

## 5 Numerical Results

In this section, the instances created to perform numerical analyses are detailed. Then, the numerical value of the algorithmic parameters are given and we explain

how they are obtained. Afterwards, algorithmic and managerial insights obtained from numerical experiments are provided.

## 5.1 Instances

Each instance is composed of a graph containing the geographic characteristics of the patients and of the hospital, and of additional information about the patients, the pharmacists, and the nurses. With the objective of creating realistic instances, interviews have been carried out in healthcare institutions in Belgium and France. The instance characteristics detailed below take into account the information gathered through those contacts.

For each graph, patient locations are generated as X-Y coordinates on a plan. The distance matrix is built using Euclidean distances rounded up to the first decimal. In each graph, 25% of the patients are concentrated in a small zone that represents a city center, 50% are located around the city center in a zone that can be seen as an urban area, and the 25% remaining are situated further away in a regional area. The hospital location is located in the urban area, outside of the city center, as it is often the case in practice.

We consider several settings for the following instance parameters: the number of patients, the length of the planning horizon, the stability times, and the time windows. The number of patients  $|\mathcal{J}|$  is equal to either 25, 50, or 100. The length of the planning horizon ( $L_0 - E_0$ ) is either 10 or 14 hours. The stability times are either short or long. Short (resp. long) stability times are generated randomly as a multiple of 30 minutes ranging from 120 to 240 minutes (resp. 300 to 480 minutes). Instances have either one of two settings: 25% of short and 75% of long stability times, or 50%-50%. Time windows are either equal to 120 minutes (short) or to 240 minutes (long). Instances have either one of two settings: 25% of short and 75% of long time windows, or 75%-25%. For all instances, the processing time of a drug is randomly generated in  $\{20, 30, 40, 60, 70, 80\}$  minutes. Finally, the administration time is randomly generated in  $\{20, 30, 40, 50, 60, 70, 80\}$  minutes. Table 2 summarizes the possible instance settings.

Parameter	Values	# Settings
Number of patients	{25, 50, 100}	3
Length of the planning horizon (hours)	{10, 14}	2
Percentage of short stability times	{25, 50}	2
Percentage of short time windows	{25, 75}	2

Table 2: Instance parameters.

In total, there exist 24 combinations of those four parameter values, eight for each size of the patient list. One graph was created for each one of those combinations. For each one of the 24 graphs, several worker configurations have been systematically generated, i.e., the number of pharmacists, the number of nurses, and the length of the working shift. The algorithm proposed in this work has been slightly modified to generate judicious values for the number of nurses and the number of pharmacists, provided a shift size ( $D^{max}$ ) of 6 or 8 hours. The resulting set of test instances contains 93 instances based on 24 different graphs. These instances are designed to be relatively difficult to solve: decreasing by one the number of pharmacists or the number of nurses may lead to infeasible instances. A folder containing the 93 instances is available at the following address: <https://hdl.handle.net/2268/292474>.

## 5.2 Algorithm configuration

The automatic configuration tool `irace` [López-Ibáñez et al., 2016] is used to configure the proposed algorithm. For this purpose, ten training instances different from the test instances, but generated according to the same principles, are used. For each algorithmic parameter, we define an initial range of possible values. Based on the list of parameters and their respective ranges, `irace` returns the final set of parameter values, called configuration, that performed best on the training instances among a large number of configurations tested through statistical races.

Table 3 summarizes the algorithmic parameters introduced in former sections by categorizing them according to the algorithmic component they belong to. For each parameter, the initial range is provided as well as the final value recommended by `irace`.

## 5.3 Experiments on test instances

The computer used for the numerical experiments is an Intel(R) Core(TM) i7-8665U CPU @1.90GHz. Five runs are performed for each instance, with a time limit per run in seconds equal to twenty times the number of customers, that is, 500, 1000, and 2000 seconds for instances of 25, 50, and 100 patients respectively.

Table 4 is divided into four groups of columns. The instance characteristics are reported in the eight first columns. The next four columns relate to the global schedule, which includes the production and administration schedules, while two additional groups of two columns are dedicated to the production schedule and to the administration schedule respectively.

For each instance, Table 4 reports the number of feasible solutions obtained over five runs, as well as the average solution value, the best solution value  $C^*$ ,

Algorithmic component	Parameter	Type	Authorized range	Final value
Objective function (4.2)	$\alpha_{min}$	integer, step 10	[10,50]	20
	$\alpha_{max}$	integer, step 10	$[\alpha_{min},1000]$	60
	$\mu$	real, 1 digit	[1,2]	1.4
	$\xi$	integer, step 10	[10,100]	80
Simulated annealing (4.5)	$\theta_0$	real, 3 digits	[0.010,0.050]	0.019
	$\kappa$	real, 2 digits	[0.01,0.50]	0.03
	$\eta$	real, 3 digits	[0.950,0.999]	0.965
Stopping criteria (4.5)	$\omega$	integer, step 50	[50,500]	350
	$\omega'$	integer, step 50	[50,2000]	900
Bounds on $k$ and $q$ (4.5)	$\lambda_{inner}$	real, 2 digits, step 0.05	[0.10,0.40]	0.30
	$\delta$	integer	[2,15]	11
	$\lambda_{outer}$	real, 2 digits, step 0.05	[0.10,0.40]	0.20
Randomization factors (4.6)	$v^P$	real, 1 digit, step 0.5	[1,5]	3.0
	$v^R$	real, 1 digit, step 0.5	[1,5]	4.0

Table 3: Algorithmic parameters values.

and the difference between the best and the worst solution values computed as a percentage of  $C^*$ .

The average duration of the production and of the administration schedules are shown in separate sections, both in absolute terms and as a percentage of the total available working time, which is equal to the length of the shift  $D^{max}$  multiplied by the number of pharmacists or nurses.

Instance characteristics							Global solution			Production		Administration			
	# Patients	Planning horizon (hours)	Short stability times (%)	Short time windows (%)	Shift length (hours)	# Pharmacists	# Nurses	# Feasible (5 runs)	Average duration	Best duration	Delta min-max	Average duration		Average duration	
Graph 01	25	10	25	25	8	3	5	5	3055.52	3032.9	1.7%	1240.00	86.11%	1815.52	75.65%
					8	4	4	5	3001.60	2971.8	1.6%	1240.00	64.58%	1761.60	91.75%
					6	4	5	5	2976.28	2952.5	1.5%	1240.18	86.12%	1736.10	96.45%
Graph 02	25	10	25	75	8	3	6	5	3313.32	3257.7	3.1%	1300.00	90.28%	2013.32	69.91%
					8	4	5	5	3183.64	3178.8	0.4%	1300.00	67.71%	1883.64	78.49%
					6	4	6	5	3171.80	3163.8	0.7%	1300.00	90.28%	1871.80	86.66%
Graph 03	25	10	50	25	8	3	6	4	3689.25	3648.5	2.9%	1410.00	97.92%	2279.25	79.14%
					8	5	5	5	3526.66	3482.4	2.4%	1411.86	58.83%	2114.80	88.12%
					6	4	7	5	3524.38	3492.3	2.0%	1410.00	97.92%	2114.38	83.90%
Graph 04	25	10	50	75	8	3	6	5	3514.78	3476.1	2.2%	1310.00	90.97%	2204.78	76.55%
					8	4	5	5	3312.48	3287.1	1.5%	1310.00	68.23%	2002.48	83.44%
					6	4	6	5	3324.38	3279.9	2.7%	1310.00	90.97%	2014.38	93.26%

Continued on next page

Continued from previous page

Continued from previous page															
Instance characteristics							Global solution				Production		Administration		
	# Patients	Planning horizon (hours)	Short stability times (%)	Short time windows (%)	Shift length (hours)	# Pharmacists	# Nurses	# Feasible (5 runs)	Average duration	Best duration	Delta min-max	Average duration		Average duration	
Graph 05	25	14	25	25	8	3	5	5	3336.72	3324.3	1.0%	1260.00	87.50%	2076.72	86.53%
					6	4	6	5	3329.54	3316.9	1.0%	1260.00	87.50%	2069.54	95.81%
Graph 06	25	14	25	75	8	3	5	5	3014.50	2997.9	1.1%	1050.00	72.92%	1964.50	81.85%
					6	3	6	5	3022.42	3004.5	1.3%	1052.14	97.42%	1970.28	91.22%
Graph 07	25	14	50	25	8	3	5	5	3488.34	3477.9	0.7%	1370.00	95.14%	2118.34	88.26%
					6	4	7	5	3478.78	3433.1	2.1%	1370.20	95.15%	2108.58	83.67%
Graph 08	25	14	50	75	8	3	5	5	3163.24	3145.1	2.1%	1222.28	84.88%	1940.96	80.87%
					6	4	6	5	3159.86	3140.7	1.5%	1223.46	84.96%	1936.40	89.65%
Graph 09	50	10	25	25	8	6	11	3	7272.97	7117.4	3.4%	2660.00	92.36%	4612.97	87.37%
					8	7	11	5	6944.18	6865.0	3.4%	2660.22	79.17%	4283.96	81.14%
					8	8	10	5	6864.76	6823.8	1.6%	2660.00	69.27%	4204.76	87.60%
					8	10	9	5	6839.12	6809.3	1.2%	2666.84	55.56%	4172.28	96.58%
					6	8	13	5	6797.56	6749.9	1.5%	2660.00	92.36%	4137.56	88.41%
					6	9	13	5	6758.86	6720.6	1.2%	2660.00	82.10%	4098.86	87.58%
					6	10	12	5	6735.68	6686.6	1.4%	2660.00	73.89%	4075.68	94.34%
Graph 10	50	10	25	75	8	7	8	5	6311.08	6276.7	1.2%	2730.00	81.25%	3581.08	93.26%
					6	9	10	5	6162.52	6133.3	1.2%	2730.00	84.26%	3432.52	95.35%
Graph 11	50	10	50	25	8	6	10	5	6723.62	6649.6	3.0%	2499.50	86.79%	4224.12	88.00%
					8	7	10	5	6472.50	6443.6	0.9%	2493.32	74.21%	3979.18	82.90%
					8	8	9	5	6591.22	6574.7	0.4%	2532.64	65.95%	4058.58	93.95%
					6	7	12	3	6499.90	6449.6	1.3%	2490.00	98.81%	4009.90	92.82%
					6	8	12	5	6445.72	6391.3	1.8%	2496.52	86.68%	3949.20	91.42%
Graph 12	50	10	50	75	8	6	9	5	6317.40	6239.9	2.4%	2450.00	85.07%	3867.40	89.52%
					8	7	9	5	6155.98	6101.2	2.1%	2450.04	72.92%	3705.94	85.79%
					8	9	8	5	6061.40	6006.8	1.6%	2454.16	56.81%	3607.24	93.94%
					6	7	11	5	6212.64	6171.8	1.5%	2450.00	97.22%	3762.64	95.02%
					6	8	11	5	6115.14	6064.4	1.4%	2450.00	85.07%	3665.14	92.55%
Graph 13	50	14	25	25	8	6	8	5	6287.18	6254.9	1.2%	2610.92	90.66%	3676.26	95.74%
					6	8	11	5	6301.22	6254.9	1.2%	2616.92	90.87%	3684.30	93.04%
					6	9	11	5	6250.62	6216.0	1.2%	2610.02	80.56%	3640.60	91.93%
Graph 14	50	14	25	75	8	6	8	5	6309.14	6289.8	0.7%	2641.96	91.73%	3667.18	95.50%
					6	8	11	5	6301.50	6252.7	1.5%	2640.56	91.69%	3660.94	92.45%
					6	9	11	5	6325.78	6246.0	2.0%	2665.32	82.26%	3660.46	92.44%
Graph 15	50	14	50	25	8	5	9	5	6176.24	6145.6	0.8%	2350.00	97.92%	3826.24	88.57%
					8	6	9	5	6176.50	6132.9	1.6%	2350.28	81.61%	3826.22	88.57%
					6	7	11	4	6146.55	6108.5	1.1%	2350.00	93.25%	3796.55	95.87%
					6	8	11	4	6166.03	6136.8	0.8%	2350.00	81.60%	3816.03	96.36%
Graph 16	50	14	50	75	8	6	9	5	6398.64	6383.2	0.7%	2591.84	89.99%	3806.80	88.12%
					8	7	9	5	6414.90	6337.0	2.5%	2611.10	77.71%	3803.80	88.05%
					6	8	12	5	6306.12	6230.7	2.0%	2593.14	90.04%	3712.98	85.95%
					6	9	11	4	6456.33	6402.0	1.2%	2617.73	80.79%	3838.60	96.93%
Graph 17	100	10	25	25	8	12	17	5	11947.68	11892.9	1.0%	4620.00	80.21%	7327.68	89.80%
					8	14	16	4	11921.83	11821.6	2.0%	4648.98	69.18%	7272.85	94.70%
					6	14	28	5	11552.78	11527.4	0.3%	4620.00	91.67%	6932.78	68.78%

Continued on next page

Continued from previous page

Continued from previous page															
Instance characteristics							Global solution				Production		Administration		
	# Patients	Planning horizon (hours)	Short stability times (%)	Short time windows (%)	Shift length (hours)	# Pharmacists	# Nurses	# Feasible (5 runs)	Average duration	Best duration	Delta min-max	Production		Administration	
												Average duration	Average duration	Average duration	Average duration
Graph 18	100	10	25	75	6	16	21	5	11507.62	11423.0	1.1%	4620.00	80.21%	6887.62	91.11%
					6	23	20	5	11415.34	11392.4	0.5%	4620.00	55.80%	6795.34	94.38%
					8	14	18	4	11861.00	11827.7	0.6%	5010.00	74.55%	6851.00	79.29%
					8	15	18	1	11842.70	11842.7	0.0%	5010.00	69.58%	6832.70	79.08%
					8	17	15	4	11876.33	11803.9	1.3%	5010.93	61.41%	6865.40	95.35%
					6	16	35	2	11712.50	11642.5	1.2%	5010.00	86.98%	6702.50	53.19%
Graph 19	100	10	50	25	6	18	32	4	11588.03	11571.8	0.4%	5010.00	77.31%	6578.03	57.10%
					6	31	19	3	11647.67	11538.3	1.5%	5020.80	44.99%	6626.87	96.88%
					8	13	20	5	12447.82	12411.9	0.7%	4900.08	78.53%	7547.74	78.62%
					8	14	19	5	12402.20	12333.9	1.3%	4903.90	72.97%	7498.30	82.22%
					6	16	22	5	12386.20	12317.7	1.6%	4900.00	85.07%	7486.20	94.52%
					6	17	22	5	12215.82	12110.7	1.6%	4900.24	80.07%	7315.58	92.37%
Graph 20	100	10	50	75	8	14	18	4	12652.05	12588.0	0.8%	5178.45	77.06%	7473.60	86.50%
					8	15	17	5	12446.46	12373.1	0.9%	5141.96	71.42%	7304.50	89.52%
					6	16	32	4	12213.30	12185.4	0.4%	5120.03	88.89%	7093.28	61.57%
					6	17	32	5	12267.20	12206.8	1.0%	5131.36	83.85%	7135.84	61.94%
					6	18	22	5	12321.56	12225.0	1.5%	5134.78	79.24%	7186.78	90.74%
					8	11	17	5	12372.10	12203.9	3.2%	4982.16	94.36%	7389.94	90.56%
Graph 21	100	14	25	25	8	12	17	5	12279.94	12228.7	1.2%	4984.22	86.53%	7295.72	89.41%
					6	19	22	5	12223.80	12171.5	1.1%	4997.96	73.07%	7225.84	91.24%
					6	20	21	4	12175.50	12085.4	1.2%	5005.50	69.52%	7170.00	94.84%
					8	11	16	4	11468.75	11420.7	0.8%	4946.05	93.68%	6522.70	84.93%
					8	12	16	5	11348.88	11302.6	0.9%	4951.70	85.97%	6397.18	83.30%
					8	13	14	4	11486.53	11324.2	2.3%	4991.18	79.99%	6495.35	96.66%
Graph 22	100	14	25	75	8	14	14	4	11612.03	11448.5	3.5%	5053.23	75.20%	6558.80	97.60%
					6	16	26	5	11524.28	11452.0	1.0%	5003.78	86.87%	6520.50	69.66%
					6	17	20	5	11507.74	11345.0	3.0%	5027.00	82.14%	6480.74	90.01%
					6	18	19	3	11536.63	11460.2	1.2%	5005.47	77.24%	6531.17	95.48%
					8	12	17	5	13125.34	12858.0	3.7%	5283.78	91.73%	7841.56	96.10%
					8	13	17	5	13061.90	13013.5	1.2%	5274.06	84.52%	7787.84	95.44%
Graph 23	100	14	50	25	6	16	23	5	12905.26	12861.5	0.7%	5269.46	91.48%	7635.80	92.22%
					6	17	23	5	12800.42	12769.0	0.6%	5251.88	85.82%	7548.54	91.17%
					8	11	18	5	12694.04	12604.2	1.6%	4836.72	91.60%	7857.32	90.94%
					8	12	18	5	12839.56	12736.3	1.7%	4838.86	84.01%	8000.70	92.60%
Graph 24	100	14	50	75	8	16	17	1	12884.90	12884.9	0.0%	5051.50	65.77%	7833.40	96.00%
					6	24	29	3	13265.03	13125.8	3.0%	5162.90	59.76%	8102.13	77.61%
					6	25	29	4	13040.90	12896.6	2.0%	5224.05	58.05%	7816.85	74.87%
					8	11	18	5	12694.04	12604.2	1.6%	4836.72	91.60%	7857.32	90.94%

Table 4: Results for benchmark instances.

## 5.4 The importance of the LP component

As explained in Section 4.4, the proposed solution method relies on an LP procedure to determine optimal start times for both the production and administration sequences. This computation takes place each time a solution is accepted in the inner loop (See Algorithm 1, Line 17).

The usefulness of this LP component has been challenged by running experiments with different triggers. The results are found in Table 5. They are aggregated by instance size and show the average number of feasible solutions found over five runs as well as the percentage time spent in the LP procedure.

The original trigger, "when a solution is accepted in the inner loop", is denoted *Accept*. The considered alternatives are described below.

- *Shaking*: After each shaking (on the initial solution of the inner loop), and whenever a new best solution is found.
- *New best*: Whenever a new best solution is found.
- *Post-opt*: Only as a post-optimization step on the final solution.
- *Threshold*: If an accepted solution does not deteriorate excessively the objective function value compared to the one of the incumbent solution of the inner loop. A threshold deterioration  $\nu \in \{0\%, 1\%, 2\%\}$  is set. Note that, as it is the case to detect improvements, the original objective function is used.
- *Time slack*: Considering an accepted solution, when  $\max_{r \in \mathcal{R}} (L(r) - E(r))$  is relatively large ( $\geq \psi$  minutes). If this criteria is met, it indicates that the opportunity for timing optimization is non negligible. We carried experiments for  $\psi \in \{2, 5, 10\}$ .

The *Threshold* and *Time slack* triggers restrain the set of accepted solutions that are reoptimized using the LP procedure. *Threshold* focuses on reoptimizing those candidate solutions that are more likely to become the new incumbent of the inner loop. *Time slack* concentrates on candidate solutions whose current schedule might be very different from the optimal one. Both the *Threshold* and the *Time slack* triggers must be parameterized. In both cases, three settings were considered as can be seen above in the trigger descriptions. Only the option yielding the best results in terms of feasibility is shown in Table 5.

It can be observed that the time spent in the LP component for the original trigger is considerable for small instances while it becomes more reasonable as the instance size increases. When comparing the results for the *Accept* and *Post-opt* trigger, it is clear that the LP procedure increases the number of feasible solutions found over five runs by the proposed algorithm.



	$ \mathcal{J} $	<i>Accept</i>	<i>Shaking</i>	<i>New best</i>	<i>Post-opt</i>	<i>Threshold 2%</i>	<i>Time slack 5'</i>
Average #	25	4.95	4.95	5.00	4.70	4.90	4.90
feasible	50	4.79	4.60	4.58	3.39	4.76	4.79
solutions	100	4.30	4.05	3.98	3.03	4.28	3.13
Time spent	25	77.87%	49.78%	0.38%	0.00%	77.41%	1.82%
in LP (%)	50	46.27%	12.48%	0.32%	0.00%	47.06%	0.33%
	100	28.71%	4.99%	0.23%	0.00%	28.59%	0.01%

Table 5: Comparison of different triggers for the LP component.

For instances with 25 customers, the *New best* trigger slightly outperforms the *Accept* trigger both in terms of feasibility and average cost of the obtained solutions. Also, the time spent by the LP is extremely small for the *New best* trigger. Unfortunately, the quality of the results decrease very significantly with the instance size. Conclusions are the same regarding the *Shaking* trigger, even if the time spent in the LP procedure is more important than the one of the *New best* trigger.

For the *Threshold* trigger, the best results were obtained with a 2% deterioration threshold, which is in fact nearly equivalent to the *Accept* trigger, since very few accepted solutions exceed this threshold. This explains the similarity of the results for those two triggers. Since decreasing the threshold decreases the quality of the results, especially on large instances, this trigger is not a valid alternative.

The case of the *Time slack* trigger is not straightforward. The best results were obtained on small instances with a 5' time slack, and the quality of the results was even slightly improved:  $-0.38\%$  and  $-0.11\%$  on the objective function value for instances with 25 and 50 customers respectively. However, for instances with 100 customers, the quality of the results decrease drastically. Nine instances could not be solved at all after five runs. This was due to the fact that very few solutions satisfied the trigger, resulting in a case that is very similar to the *Post-opt* trigger. In fact, for many large instances, even a very small value of  $\psi$  would not result in an improvement since the trigger is difficult to satisfy. Consequently, more work would be required to determine if there exists an interesting time-related trigger.

## 5.5 The cost of integration

Table 6 compares the duration obtained for the integrated problem under consideration with the duration that would be obtained when production and administration operations are planned sequentially.

Two conditions must mandatorily be met to ensure that the production and administration subproblems can be considered independently. First, stability constraints must be relaxed, and second, planning horizons must not overlap. The first condition alone is not sufficient since trip earliest start times are bounded by production completion times.

Clearly, having to integrate the production and administration subproblems comes at a cost, but it is a necessary condition to be able to take stability constraints into account and to suppress the drug inventories between production and administration operations. In Table 6, this cost of integration is expressed as a percentage increase of the duration of the non-integrated schedules. For the production schedule without any integration, the obtained duration is always equal to the optimum solution, that is, the sum of the processing times. For the administration schedule, the average duration obtained over five runs is considered as the reference value. Note that, in order to obtain those values, experiments have been sped up by dividing the running time by five since the algorithm converges much faster when the production and the administration subproblems are disconnected.

The results are aggregated based on different values of the instance parameters. It can clearly be seen that, for all groups of instances, the production duration of the integrated solution is very close to the sum of the processing times. Indeed, the production schedules obtained for the integrated problem contain nearly no idle time. However, the total duration of the administration operations increases on average by approximately 15%. The multi-trips participate to this cost increase. For the integrated problem, solutions contain around two trips per nurse on average over all instances, while they never occur when the subproblems are treated sequentially. Increasing the length of the planning horizon reduces the magnitude of the cost increase, and so does increasing the proportion of long stability times. This can easily be explained by the fact that those two factors influence the level of integration of the two subproblems. Note that the stability constraints may narrow dynamic time windows more substantially if static time windows are large. This may explain that instances with 25% of short time windows yield a slightly more important cost of integration than instances with 75% of short time windows.

## 5.6 The impact of stability

Experiments were conducted on instances with 25 and 50 patients to compare the solutions with and without stability constraints. In practice, the stability of

		Global duration incr.	Production duration incr.	Administration duration incr.
All instances		+8.39%	+0.49%	+14.47%
Planning horizon	10 hours	+8.94%	+0.12%	+16.90%
	14 hours	+7.72%	+0.19%	+11.43%
Stability	25% short	+7.23%	+0.07%	+13.21%
	50% short	+9.56%	+0.19%	+15.89%
Time windows	25% short	+8.69%	+0.11%	+15.36%
	75% short	+8.08%	+0.17%	+13.74%

Table 6: The cost of integration.

some drugs may be larger than the planning horizon and hospitals may prefer to administer only these at home in order to ease the planning process. However, providing hospitals with suitable decision-making tools is necessary to enlarge the set of patients that can benefit from home hospitalization.

		Global duration incr.	Production duration incr.	Administration duration incr.
All instances		+4.07%	+0.14%	+6.89%
Stability	25% short	+3.06%	+0.07%	+5.23%
	50% short	+4.97%	+0.19%	+8.37%
Planning horizon	10 hours	+2.73%	+0.10%	+4.55%
	14 hours	+5.96%	+0.19%	+10.18%

Table 7: The cost of stability.

Table 7 shows the duration increase when stability is taken into account. Note that in our experiments, the solutions obtained when stability is relaxed contain no production idle time. This is expected since the optimal solutions of this relaxed problem never contain idle time. Indeed, any feasible solution containing production idle time may be transformed into a solution that does not have any, by pushing all production operations backward in time. In the absence of stability times, such a strategy may only enlarge the dynamic routing time windows, possibly resulting in a decrease of the routing cost. Consequently, the production duration increase reported in Table 7 can also be interpreted as the total production idle time as a percentage of the total processing time.

Table 7 also shows a significant increase of the total administration time in the presence of stability constraints. As shown in Section 4.4, a drug with a short

stability time may shrink the corresponding dynamic administration time window, and significantly impact the administration schedule. The cost increase is higher for instances that have originally a larger number of short stability times (120 to 240 minutes) and for instances with a longer planning horizon, since stability constraints amplify the level of integration more significantly in those cases.

## 5.7 The impact of time windows

Experiments were performed on instances with 25 and 50 patients by setting all time window lengths to identical values, with the objective of evaluating the relation between time window lengths and the total cost. First, all time windows have been relaxed, i.e., considered to have a duration equal to the planning horizon. Second, the following values have been set as time window size for all the patients: 480, 360, 240, and 120 minutes. The time window centers are kept unchanged compared to those of the original instances. The total duration as well as the production and administration duration values are reported in Table 8 in terms of their increase compared to the case where time windows are relaxed. The two last columns show the administration duration increase separately for instances having 25% and 50% of short stability times.

TW size	Global duration incr.	Production duration incr.	Administration duration incr.	Administration duration	
				25% short stability	50% short stability
480	+0.55%	+0.00%	+0.95%	+0.88%	+1.02%
360	+1.23%	+0.00%	+2.08%	+1.68%	+2.47%
240	+2.16%	+0.03%	+3.66%	+3.04%	+4.29%
120	+5.73%	+0.20%	+9.61%	+7.77%	+11.45%

Table 8: Cost increase compared to instances with relaxed time windows.

Note that, when time windows are relaxed, an optimal solution without idle production time is not guaranteed, contrarily to the case of relaxed stability times. We illustrate this point using a toy example where two patients need a similar drug with  $ST_1 = ST_2 = 150$ ,  $P_1 = P_2 = 40$ , and  $S_1 = S_2 = 30$ . Travel times are such that  $T_{01} = T_{02} = T_{10} = T_{20} = 80$ . There is one pharmacist and one nurse available. Clearly, it is not possible to produce the two drugs and then travel to any of those two patients before the stability time of the first produced drug expires ( $P_1 + P_2 + T_{01} = 40 + 40 + 80 > 150$ ). The only possible solution is to administer each drug in a specif trip. Without loss of generality, say that the production of the first drug starts at time 0. The second patient may be reached the earliest at

$P_1 + T_{01} + S_1 + T_{10} + T_{02} = 310$ . Consequently, the production start time of the second drug may not start before 160, resulting in a minimum idle time of 120 time units.

In the considered instances, the production idle times are anecdotal but not necessarily equal to zero when time windows are relaxed. It can be observed that the decrease of the time window size only results in a very small amount of supplementary production idle time, while the administration time gradually increases with the reduction of the time window size. The increase is strongly amplified as time window sizes become more restrictive. The cost increase is more important for instances with short stability times as can be seen in the last two columns on Table 8.

## 6 Conclusion

In this work, we presented an integrated production scheduling and vehicle routing problem encountered in the context of home chemotherapy. The problem is composed of two subproblems: the scheduling of drug production operations, which are performed by pharmacists in the hospital, and the routing and scheduling of drug administration operations, which are performed by a set of nurses and take place at the patients' homes.

The studied problem includes several constraints related to the timing of the production and administration operations. Firstly, a drug stability time limits the time span allowed between the drug production and administration. Secondly, the administration of drugs are subject to time windows. We showed how those timing constraints, as well as the presence of a joint planning horizon, causes the two subproblems to be closely intertwined. We elaborated heuristic and exact algorithmic components that specifically take into consideration this interdependence.

We created realistic test instances for the studied problem and configured the proposed solution method using `irace`, an automatic configuration tool.

A linear program was repeatedly used inside the large neighborhood search in order to reoptimize the schedules, provided sequences obtained using large neighborhood search moves. We demonstrated that this component was crucial to obtain feasible solutions, especially for larger instances, and we believe that further work on the criteria that trigger the call to this linear program may benefit the effectiveness and efficiency of solution methods.

We performed numerical experiments and evaluated the cost of having to integrate the production and administration operations as opposed to a situation where the two plannings would be established separately. Our focus here was to provide insights about the effects of problem characteristics that are not compatible with a sequential planning approach.

Finally, we evaluated the impact of time windows and stability constraints. The provided insights may be useful when setting up a home chemotherapy process, in order to raise awareness about the impact of patient-related characteristics on operational decisions.

## References

- Y. Adulyasak, J.-F. Cordeau, and R. Jans. The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, 55:141 – 152, 2015.
- R. Armstrong, S. Gao, and L. Lei. A zero-inventory production and distribution problem with a fixed customer sequence. *Annals of Operations Research*, 159(1):395–414, 2008.
- K. Braekers, R. F. Hartl, S. N. Parragh, and F. Tricoire. A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience. *European Journal of Operational Research*, 248(2):428 – 443, 2016.
- 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.
- H.-K. Chen, C.-F. Hsueh, and M.-S. Chang. Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research*, 36(7):2311–2319, 2009.
- Z.-L. Chen. Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research*, 58(1):130–148, 2010.
- Z.-L. Chen and G. Vairaktarakis. Integrated scheduling of production and distribution operations. *Management Science*, 51(4):614–628, 2005.
- 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.
- M. Drexl. Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3):297–316, 2012.
- European Commission. Joint report on health care and long-term care systems & fiscal sustainability, 2016.

- V. François, Y. Arda, Y. Crama, and G. Laporte. Large neighborhood search for multi-trip vehicle routing. *European Journal of Operational Research*, 255(2): 422–441, 2016.
- V. François, Y. Arda, and Y. Crama. Adaptive large neighborhood search for multi-trip vehicle routing with time windows. *Transportation Science*, 53(6): 1706–1730, 2019.
- J. Garcia, S. Lozano, and D. Canca. Coordinated scheduling of production and delivery from multiple plants. *Robotics and Computer-Integrated Manufacturing*, 20(3):191–198, 2004.
- H. Geismar, G. Laporte, L. Lei, and C. Sriskandarajah. The integrated production and transportation scheduling problem for a product with a short lifespan. *Inform's Journal on Computing*, 20(1):21–33, 2008.
- I. Karaođlan 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.
- 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.
- 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):1–24, 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.
- F. Lehuédé, P. Grangier, A. Grimault, and R. Masson. Efficient handling of synchronization in three vehicle routing problems. In *Odysseus 2015*, Odys, Ajaccio, France, 2015.
- M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, , M. Birattari, and T. Stütze. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43 – 58, 2016.
- R. Lowenthal, A. Piaszczyk, G. Arthur, and S. O'Malley. Home chemotherapy for cancer patients: cost analysis and safety. *The Medical journal of Australia*, 165 (4):184–187, 1996.

- R. Masson, F. Lehuédé, and O. Péton. Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers. *Operations Research Letters*, 41(3):211 – 215, 2013.
- R. Masson, F. Lehuédé, and O. Péton. The dial-a-ride problem with transfers. *Computers & Operations Research*, 41:12 – 23, 2014.
- 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.
- Y. Nagata, O. Bräysy, and W. Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724–737, 2010.
- A. Olivera and O. Viera. Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research*, 34:28–47, 2007.
- M. S. Rasmussen, T. Justesen, A. Dohn, and J. Larsen. The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3):598 – 610, 2012.
- A. Robbes, Y. Kergosien, V. André, and J.-C. Billaut. Efficient heuristics to minimize the total tardiness of chemotherapy drug production and delivery. *Flexible Services and Manufacturing Journal*, in press, published online October 10, 2021.
- S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40: 455–472, 2006.
- M. Savelsbergh. The vehicle routing problem with time windows : minimizing route duration. *ORSA Journal on Computing*, 4(2):146–154, 1992.
- P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher and J.-F. Puget, editors, *Principles and Practice of Constraint Programming — CP98: 4th International Conference, CP98 Pisa, Italy, October 26–30, 1998 Proceedings*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431. Springer, Berlin, Heidelberg, 1998.
- R. Turkeš, K. Sörensen, and L. M. Hvattum. Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search. *European Journal of Operational Research*, 292(2):423–442, 2021.



- C. A. Ullrich. Integrated machine scheduling and vehicle routing with time windows. *European Journal of Operational Research*, 227(1):152 – 165, 2013.
- T. Vidal, T. Crainic, M. Gendreau, and C. Prins. Time-window relaxations in vehicle routing heuristics. *Journal of Heuristics*, 21(3):329–358, 2015.
- C. Viegutz and S. Knust. Integrated production and distribution scheduling with lifespan constraints. *Annals of Operations Research*, 213(1):293–318, 2014.
- 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, 2019.