






Article

Outpatient Appointment Optimization: A Case Study of a Chemotherapy Service

Quoc Nhat Han Tran ^{1,2,*} , Nhan Quy Nguyen ^{2,3} , Hicham Chehade ¹ , Lionel Amodeo ² 
and Farouk Yalaoui ^{2,3} 

¹ Opta LP S.A.S., 2 Rue Gustave Eiffel, 10430 Rosieres-pres-Troyes, France; chehade@opta-lp.com

² Computer Sciences and Digital Society Laboratory (LIST3N), Université de Technologie de Troyes, 12 Rue Marie Curie, CS 42060, 10004 Troyes, France; nhan_quy.nguyen@utt.fr (N.Q.N.); lionel.amodeo@utt.fr (L.A.); farouk.yalaoui@utt.fr (F.Y.)

³ Chaire Connected Innovation, Université de Technologie de Troyes, 12 Rue Marie Curie, 10004 Troyes, France

* Correspondence: quoc_nhat_han.tran@utt.fr

Featured Application: This study provides a procedure-detailed scientific approach for the outpatient planning problem at the chemotherapy department.

Abstract: In this paper, we study a complex outpatient planning problem in the chemotherapy department. The planning concerns sequences of patients' treatment sessions subject to exact in-between resting periods (i.e., exact time-lags). The planning is constrained by the hospital infrastructure and the availability of medical staff (i.e., multiple time-varying resources' availability). In order to maximize the patients' service quality, the objective of the function considered is to minimize the total wait times, which is equivalent to the criteria for minimizing the total completion time. Our main contribution is a thorough analysis of this problem, using the Hybrid Flow Shop problem as a theoretical framework to study the problem. A novel Mixed Integer Linear Programming (MILP) is introduced. Concerning the resolution methods, priority-based heuristics and an adapted genetic algorithm (GA) are presented. Numerical experiments are conducted on historical data to compare the performances of the approximate resolution methods against the MILP solved by CPLEX. Numerical results confirm the performances of the proposed methods.

Keywords: outpatient; planning techniques; hybrid flow shop; time-lag; time-varying resources; genetic algorithm; dispatching rules



Citation: Tran, Q.N.H.; Nguyen, N.Q.; Chehade, H.; Amodeo, L.; Yalaoui, F. Outpatient Appointment Optimization: A Case Study of a Chemotherapy Service. *Appl. Sci.* **2022**, *12*, 659. <https://doi.org/10.3390/app12020659>

Academic Editor: Jing Jin

Received: 17 November 2021

Accepted: 4 January 2022

Published: 10 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

According to Hesaraki [1], there are two treatment settings in hospitals: *inpatient* care and *outpatient* care. Inpatient care requires more than one day of hospitalization. Outpatient care concerns patients who are admitted to and leave hospitals on the same day. While inpatient care demands dedicated resources such as a bed or a room with equipment for a single patient, outpatient care allows for reusing the same resources for several patients a day, thus increasing the utility and the flexibility of the hospital infrastructure. A challenge arises with the growing number of patients' demands: ensuring the service capability of human and infrastructural resources while maintaining the quality of services.

In this work, we study the outpatient planning problem under medical resources. The hybrid flow shop problem is applied as our theoretical framework for the problem studied. The case-study context and data are provided by the service of Outpatient Care of Pneumology–Hematology–Oncology (Troyes Hospital Center, France).

The main contributions of this work are as follows: (1) a most detailed procedural analysis, to the best of our knowledge, of a chemotherapy planning problem; (2) a novel mixed-integer linear programming model that takes into account the exact time-lag and time-varying resource constraints; (3) adapted heuristics and a genetic algorithm allowing

for parallel computation to reduce time computation proposed to efficiently solve the problem studied.

This paper is organized as follows: Section 2 provides a detailed description of the problem following by a complexity analysis. Next, Section 3 gives the overview of the outpatient scheduling/planning, especially the chemotherapy. The Mixed Integer Linear Programming (MILP) model from our previous work [2] is detailed in Section 4. Section 5 is dedicated to resolution methods: eight priority-based heuristics and a genetic algorithm. Section 6 specifies the test protocol and the numerical results to gauge the performances of different approaches. Finally, Section 7 concludes this paper with our remarks and states our future works.

2. Problem Description

This section provides a detailed description of the problem followed by a classification of the problem and a complexity analysis.

2.1. Outpatient Scheduling Problem

The outpatient scheduling at the chemotherapy service includes two subproblems, according to Condotta and Shakhlevich [3]. In the first problem, in which the time horizon is considered by the number of days, the scheduler must decide the starting time of a sequence of *treatment sessions* (as known as *appointments*) and *rest days* in-between (Figure 1). The second planning problem has the time horizon limited by a day and consists of a four-stage process requiring multiple resources at each stage (Figure 2).

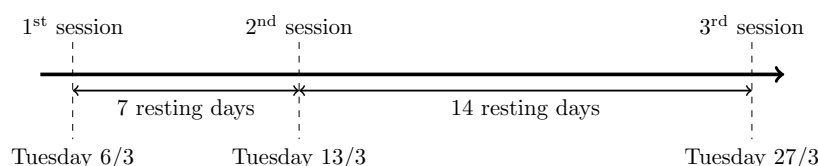


Figure 1. Example of a cycle of three appointments.

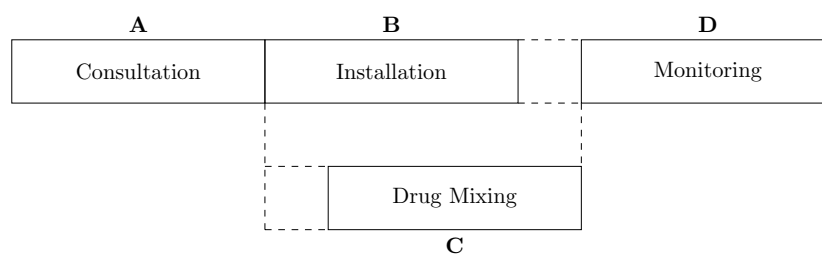


Figure 2. Gantt diagram of a chemotherapy outpatient care.

The first subproblem considers, for each patient, a complete chemotherapy outpatient therapy, which is called the *multi-day pattern*. The therapy comprises several treatment sessions (i.e., appointments) distanced by predefined resting time windows for healing, as illustrated in Figure 1. These inter-appointment periods are fixed so that the patient’s health recovers for the next session. For the sake of brevity, this sequence is denoted as a *cycle* of a patient.

The second subproblem handles the scheduling on the day of the appointment, decided by the first subproblem. After being admitted, the patient consults a doctor to check up on her/his current health condition. If the patient is judged to be not healthy enough to receive the treatment, the current session is rescheduled to another day. Otherwise, the patient follows a nurse to an available seat (a bed or an armchair). In parallel, the required chemotherapy drug is prepared at the pharmacy unit. Once the drug is ready, the patient proceeds to receive the treatment under the monitoring of a nurse.

However, in some exceptional cases, the treatment session does not require a consultation in the same day because this consultation can be made in advance. In this case, the consulting time of the patient is considered zero. Furthermore, occasionally, the mixing drug step is permitted to proceed *one day in advance* so that the waiting time can be reduced. The Gantt diagram in Figure 3 illustrates those operations.

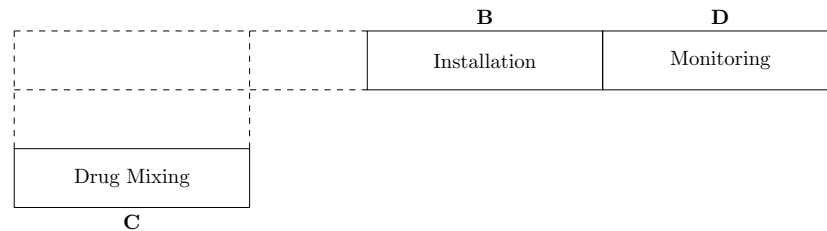


Figure 3. Gantt diagram of a particular case: no consultation, and drug mixing is allowed to process in advance.

Both subproblems have time-varying resource availability, illustrated as in Figure 4. Since the processing of each stage requires one or multiple resources, the variation in resources implies the maximal number of patients that the service can handle simultaneously. In this problem, four types of resources are considered:

- **Pharmacy:** a resource only available during the predefined opening hours. The capacity of the pharmacy is considered unlimited.
- **Doctor:** a discrete time-varying resource. A doctor can receive a maximum of one patient at a time.
- **Nurse:** a discrete time-varying shared resource. Each nurse can install a patient to a seat or simultaneously monitor at most Ψ patients.
- **Seat:** a constant renewable resource. Each patient occupies a seat from the beginning of installation until the end of the session of care.

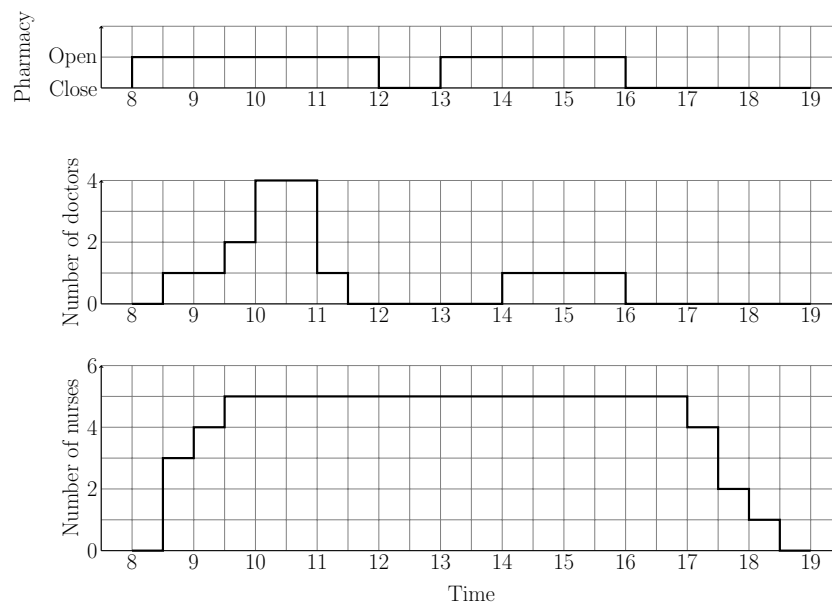


Figure 4. Example of resources' variation in a day.

2.2. Problem Categorization

We denote the problem using the Graham notation $\alpha | \beta | \gamma$ [4].

For each session, the patient should go through four stages, as mentioned earlier (Figures 2 and 3), and then, the problem belongs to the family of **Flow Shop problems**.

In addition, according to the capacity of different types of resources, the last stage (*monitoring*) can be executed for multiple patients in parallel. The problem is classified to be a **Hybrid Flow Shop** scheduling problem. Since each care session has four operations, we denote $\alpha = FH4$.

The resource constraints are denoted as $res\ \lambda\sigma\rho$ according to Blazewicz et al. [5], with new modifications introduced by Weglarz [6]. There are four types of resources required for the execution of jobs ($\lambda = 4$). No bound on the resource capacities is known a priori ($\sigma = \cdot$). The requirement of resources is bounded correspondingly to the stage ($\rho = 1$): one doctor for *consultation*, one pharmacy unit for *drug mixing*, one nurse and one chair for *installation*, and $1/\Psi$ nurse (in terms of capacity Ψ) and one chair for *monitoring*. The resource constraints notation is then $res\ 4 \cdot 1$.

The final job constraint is the waiting time between two treatment sessions of a patient (Figure 1). An outpatient chemotherapy cycle requires the patient in question to rest a certain number of days before entering the next session. In the literature, this constraint is described as the time-lag constraints, which are denoted as $l_{jj'}$ [7,8]. The time-lag is a generalization of the precedence constraint where one job has to wait an amount of time before processing, depending on another job. Due to the technical aspects of the chemotherapy care, we are only interested in the number of days between two consecutive jobs.

The objective of our studied problem is to maximize the quality of service. In this case, we tend to reduce the mean waiting time of the patients, which is equivalent to minimizing the total completion time [9].

In summary, the problem is noted as:

$$FH4 | l_{jj'}, res\ 4 \cdot 1 | \sum C_j$$

2.3. Problem Complexity

By assuming that the input is only patients with a single session and that **consultation** and **installation** time are zero, the considered problem without resources constraints is reduced to the *two stages Flow Shop* problem, which is **NP-hard** [10]. Furthermore, the resource variation is usually seen as an element that makes the planning/scheduling problem harder to solve [11]. Our problem thus should be at least **NP-hard**.

3. Literature Review

Healthcare services have been receiving a growing number of papers covering several major domains in response to many complex problems from the increasing patients' demands under limited and scarce resources. Multiple reviews [12–19] have shown an extensive view about typical challenges regarding multiple aspects, levels, and criteria.

The hybrid approach of the branch-and-bound algorithm and the constraint programming technique in [20] proposes exact solutions for the generic integrated problem of employee timetabling and job-shop scheduling. The work is then applied to the outpatient chemotherapy scheduling problem, yielding significant results. The work of Condotta and Shakhlevich [3] considers the recovery duration after each session (multi-day time-lag) and the intervention during the monitoring step (intraday time-lag) by resolving two ILP models corresponding to multi-day and intraday subproblems over the rolling horizon. Hahn-Goldberg et al. [21], on the other hand, proposed resolution methods based on Constraint Programming and MIP that yield applicable performance. Liang and Turkcan [22] tackled it using a multi-objective optimization, providing a helpful spreadsheet decision-making solution for strategical clinic management. The branch-and-price algorithm of Legrain et al. [23] finds an application in the nurse rostering problem. Under the assumptions that the pharmacy unit and oncologists are abundant, the MILP model by Hesarakı [1] focuses primarily on the last two stages because efficiently allocating nurses in a day is more critical and complex. Then, the method aggregates the time-lag progressively by applying a rolling horizon and schedule template to construct the long-term

planning. Overall, few papers consider the problem with multiple types of resources and time-varying resource availability.

On the other hand, a quick and approximate solution’s applicability in practical cases has attracted the development of heuristic and meta-heuristic. Menting and Menting [24] built several heuristic algorithms to enhance the schedule of affecting resources (nurses and seats) to patients after applying LPT rule combined with First Fit Strategy. The resolution time is less than several seconds, resulting in a good planning tool for the local hospital in daily usage. The review of Marynissen and Demeulemeester [19] lists multiple meta-heuristic approaches for multi-appointment planning and, in particular, chemotherapy scheduling. The latter invokes multiple resources and requires repeating treatment sessions over a long horizon.

As far as we know, there is a lack of work in the literature that studies all four mentioned stages. As indicated in Table 1, while some work considers the multi-day subproblem (time-lag), few include all stages (intra-day subproblem) and required resources. That is the reason why our previous work’s model [2] is the most detailed in terms of care operations so far, according to our knowledge.

Table 1. Positioning our research (in bold) among the most related ones.

Researches	Multiday (with Time-Lag)	Stages	Resources		Criteria
			Types	Variability	
Turkcan et al. [25]	Yes	4 (as a whole)	2	No	Overtime + Delay
Hahn-Goldberg et al. [21]	No	4	4	No	Makespan
Condotta and Shakhlevich [3]	Yes	2	2	Constant except middle break	Waiting, Workload
Liang and Turkcan [22]	No	1	2	Single shifts	Overtime + Waiting
Hesaraki [1]	Yes	2	2	Constant except middle break	Makespan + Total weighted flow time
Tran et al. [2]	Yes	4	4	Yes	Total completion time

4. Mathematical Model

The mathematical models of Hesaraki [1] and Condotta and Shakhlevich [3] deal principally with constant resources, which cannot be adapted to our problem where the resources vary over time. For this reason, we develop another formulation using *disjunctive binary variable* technique. This modeling approach brings us some benefits. The variation of resources can be wholly taken into account and can be extensible to any number of discrete resources types. Reordering operations can be made without complete modification of resources constraints. Readers are referred to the work of Nguyen et al. [26] for more details.

Let us denote the disjunctive binary variable as follows:

$$\sigma_{i,k,j,h}^o = \begin{cases} 1 & \text{if operation } o \text{ of appointment } k \text{ of patient } i \\ & \text{starts at time } h \text{ on day } j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\chi_{i,k,j,h}^o = \begin{cases} 1 & \text{if operation } o \text{ of appointment } k \text{ of patient } i \\ & \text{ends at time } h \text{ on day } j \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The detailed notations to the problem are stated in Table 2. With such a notation, we can describe the fundamental measures of a given treatment session k of patient i mathematically:

- Each operation o can be started and ended only once:

$$\sum_{j \in \mathcal{J}} \sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^o = 1 \tag{3}$$

$$\sum_{j \in \mathcal{J}} \sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^o = 1 \tag{4}$$

- The processing time $p_{i,k,j,h}^o$ of an operation o must be equal to the difference between the starting time $\sigma_{i,k,j,h}^o$ and the completion time $\chi_{i,k,j,h}^o$:

$$\sigma_{i,k,j,h}^o = \chi_{i,k,j,h+p_{i,k}^o}^o \tag{5}$$

- Given a date j , the completion indicator variable $\sum_{h' \leq h} \chi_{i,k,j,h'}^o$ checks whether the operation o is completed by the moment h . Therefore, if an operation \hat{o} succeeds o , the starting time of \hat{o} must not exceed o 's completion indicator:

$$\sigma_{i,k,j,h}^{\hat{o}} \leq \sum_{h' \leq h} \chi_{i,k,j,h'}^o \tag{6}$$

- Since $\sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^D$ indicates if the session k happens on day j , one can use the following formula to extract the index of the day when the treatment session happened:

$$j_{i,k} = \sum_{j \in \mathcal{J} \setminus \{0\}} \left(j \times \sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^D \right) \tag{7}$$

One can find the completion time of a process by a similar manner:

$$h_{i,k} = \sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} h \times \chi_{i,k,j,h}^D \tag{8}$$

From (4), (7) and (8), we can rewrite the completion time of a session k by multiplying $j_{i,k}$ the number of passing days by H the number of timeslots per day then adding up with $h_{i,k}$ the elapsed time of the treatment:

$$\begin{aligned} C_{i,k} &= H \times (j_{i,k} - 1) + (h_{i,k} - 1) \\ &= H \times \left(\sum_{j \in \mathcal{J} \setminus \{0\}} j \times \sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^D - \sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^D \right) + \\ &\quad + \left(\sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} h \times \chi_{i,k,j,h}^D - \sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^D \right) \\ &= H \times \sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} (j - 1) \times \chi_{i,k,j,h}^D + \sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} (h - 1) \times \chi_{i,k,j,h}^D \\ &= \sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} (H \times (j - 1) + (h - 1)) \chi_{i,k,j,h}^D \tag{9} \end{aligned}$$

- Given a date j , the resource usage at the time h is determined by the execution of operation o , which is calculated by the sum of the difference of the starting time indicator variable $\sigma_{i,k,j,h'}^o$ and the completion time indicator variable $\chi_{i,k,j,h'}^o$ from time 0 to h : $\sum_{h' \leq h} (\sigma_{i,k,j,h'}^o - \chi_{i,k,j,h'}^o)$.

Table 2. Notation of parameters.

Notation	Description
$O = \{A, B, C, D\}$	Four operations of an appointment (Figure 3)
$\mathcal{J} = \{0, 1, 2, \dots, J\}$	Set of days
j	Index of day
$\mathcal{H} = \{1, 2, \dots, H, H + 1\}$	Set of time slots of a day ($H + 1$ indicates the end of day)
h	Index of time
M	Number of places
Ψ	Maximal number of patients that one nurse can monitor in the same time
$D(j, h)$	Number of doctors at time h on day j
$I(j, h)$	Number of nurses at time h on day j
$F(j, h)$	1 if pharmacy opens at time h on day j , 0 otherwise
\mathcal{P}	Set of patients
i	Index of patient
$\Theta_i = \{1, 2, \dots, n_i\}$	Set of required appointments for patient i
k	Index of appointment
$p_{i,k}^o$	Duration of operation o for patient i at appointment k
$e_{i,k}^C$	Authorization of a priori drug mixing for patient i at appointment k
$\Delta_{i,k}$	Time-lag (days) of appointment k of patient i with respect to appointment $k - 1$

From these formulas, we obtain the MILP model as follows:

$$\text{Minimize } \sum_{i \in \mathcal{P}} \sum_{k \in \Theta_i} \sum_{j \in \mathcal{J} \setminus \{0\}} \sum_{h \in \mathcal{H}} (H \times (j - 1) + (h - 1)) \times \chi_{i,k,j,h}^D \tag{10}$$

Subject to:

$$\sum_{j \in \mathcal{J}} \sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^o = \sum_{j \in \mathcal{J}} \sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^o = 1 \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall o \in O \tag{11}$$

$$\chi_{i,k,j,h+p_{i,k}^o}^o = \sigma_{i,k,j,h}^o \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall o \in O, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{12}$$

$$\sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^o = \sum_{h \in \mathcal{H}} \chi_{i,k,j,h}^o \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall o \in O, \forall j \in \mathcal{J} \tag{13}$$

$$\sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^A = \sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^B = \sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^D \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall j \in \mathcal{J} \tag{14}$$

$$\sigma_{i,k,0,h}^o = \chi_{i,k,0,h}^o = 0 \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall o \in \{A, B, D\}, \forall h \in \mathcal{H} \tag{15}$$

$$\sigma_{i,k,j,h}^B \leq \sum_{h' \leq h} \chi_{i,k,j,h'}^A \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{16}$$

$$\sum_{\substack{j' \in \mathcal{J}: \\ j - e_{i,k}^C \leq j' < j}} \sum_{h \in \mathcal{H}} \sigma_{i,k,j',h}^C \geq \sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^A \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{17}$$

$$(1 - e_{i,k}^C) \sigma_{i,k,j,h}^C \leq \sum_{h' \leq h} \chi_{i,k,j,h'}^A \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{18}$$

$$\sigma_{i,k,j,h}^D \leq \sum_{h' \leq h} \chi_{i,k,j,h'}^B \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{19}$$

$$\sigma_{i,k,j,h}^D \leq \sum_{\substack{j' \in \mathcal{J}: \\ j - e_{i,k}^C \leq j' < j}} \sum_{h' \in \mathcal{H}} \chi_{i,k,j',h'}^C + \sum_{h' \leq h} \chi_{i,k,j,h'}^C \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{20}$$

$$\sum_{h \in \mathcal{H}} \sigma_{i,k+1,j+\Delta_{k+1}^i,h}^A = \sum_{h \in \mathcal{H}} \sigma_{i,k,j,h}^A \quad \forall i \in \mathcal{P}, \forall k \in \{1, \dots, n_i - 1\}, \forall j \in \mathcal{J} : j + \Delta_{k+1}^i \in \mathcal{J} \tag{21}$$

$$M \geq \sum_{i \in \mathcal{P}} \sum_{k \in \Theta_i} \sum_{h' \leq h} \left(\sigma_{i,k,j,h'}^B - \chi_{i,k,j,h'}^D \right) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{22}$$

$$D(j, h) \geq \sum_{i \in \mathcal{P}} \sum_{k \in \Theta_i} \sum_{h' \leq h} \left(\sigma_{i,k,j,h'}^A - \chi_{i,k,j,h'}^A \right) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{23}$$

$$I(j, h) \geq \sum_{i \in \mathcal{P}} \sum_{k \in \Theta_i} \sum_{h' \leq h} \left(\left(\sigma_{i,k,j,h'}^B - \chi_{i,k,j,h'}^B \right) + \left(\sigma_{i,k,j,h'}^D - \chi_{i,k,j,h'}^D \right) / \Psi \right) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{24}$$

$$F(j, h) \geq \sum_{h' \leq h} \left(\sigma_{i,k,j,h'}^C - \chi_{i,k,j,h'}^C \right) \quad \forall i \in \mathcal{P}, \forall k \in \Theta_i, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \tag{25}$$

Since we aim to reduce the total waiting time of the patient in the hospital, the objective function is the total completion time minimization (10), expressed by Equation (9). We summarize here the explication of each group of constraints:

1. **Operations:** (11) obligates any operation to start and end exactly *once* on the horizon, according to (3). (12) makes each operation *non-preemptive*, which means that, once an operation starts, it must be processed without interruption. Equation (13) makes their start and end dates the same (outpatient regime). For each session, three stages (*consultation, installation, and administration*) are on a same day (14). However, since day 0 is a dummy day to allow for mixing the drug in advance, if possible, for day 1, these three steps are prohibited on day 0 (15).
According to the order of operations, the *installation* never precedes *consultation* (16). The *mixing*, if a priori preparation is authorized ($e_{i,k}^C = 1$), can proceed at most one day before the day of treatment session (17); otherwise, it follows the end of *consultation* (18). Finally, *monitoring* starts when the patient has been fully installed (19) and the drug is available (20).
2. **Appointments:** Each patient has a set of appointments in which the time-lags have been decided beforehand in terms of days (21).
3. **Seats:** A patient occupies one seat from the beginning of *installation* until finishing the *monitoring* (22).
4. **Doctors:** One doctor takes in one patient at a time (23).
5. **Nurses:** A nurse can either install a patient into a seat or monitor a group of Ψ patients (24).
6. **Pharmacy:** Unlimited orders are allowed to process in parallel as long as the *mixing* procedures are in the pharmacy’s opening time (25).

5. Resolution Methods

Although the MILP model for our problem defined at Section 4 gives exact solutions with the solver CPLEX, the resolution time increases exponentially as the instance size grows. Thus, we seek other approaches with a more reasonable resolution time. We present in this section eight priority rules and a genetic algorithm adapted to the problem.

5.1. Dispatching Rule Heuristics

Due to the time-lag constraint, one must always ensure the exact waiting time between two patient sessions. With First Fit Strategy (FFS) [27], we can find a resource, precedence, and feasible planning for a sequence of appointments of a patient by scheduling as soon as possible while ensuring the time-lag and resources’ availability for the patient. In addition, the time-lag is only applied to the appointments of each patient. Hence, it is sufficient to represent the problem’s instance as an orderly list of patients.

Since the SPT rule is a well-known rule for $1 \parallel \sum C_j$ [28] by prioritizing the job with lower processing time, we first define the priority value *total processing time* as the sum of the entire elapsed time for all steps at all appointments.

We also define the *ideal total processing time*, a variation that takes into account the flexibility of jobs in the ideal case where we process without resource constraints:

$$\begin{aligned}
 IPT_{i,k} &: \text{ideal total processing time of appointment } k \text{ of patient } i \\
 &= p_{i,k}^A + \max\{p_{i,k}^B; (1 - e_{i,k}^C)p_{i,k}^C\} + p_{i,k}^D \\
 &= \begin{cases} p_{i,k}^A + \max\{p_{i,k}^B; p_{i,k}^C\} + p_{i,k}^D & \text{if C is not allowed to preceed} \\ p_{i,k}^A + p_{i,k}^B + p_{i,k}^D & \text{otherwise} \end{cases} \\
 \Rightarrow IPT_i &= \sum_{k \in \Theta_i} IPT_{i,k} : \text{ideal total processing time of patient } i
 \end{aligned}$$

Finally, note that, if we delay the first session by *one day*, the time-lag pushes the second session an equal amount, resulting in an increase of *two days* ($2 \times H$ time slots) in total on the objective (10). In other words, delaying the first session of patient i by one day increases the objective (10) by at least $n_i \times H$. For that reason, we see by intuition that prioritizing long requests reduces the objective substantially. Combining with the idea of *MTS (Most Total Successor)* [29], we construct hence the *RLIPT* priority value, as a lexicographical order $(n_i; IPT_i)$ so that the lengthier cycle (greater n_i) is prioritized, tie-broken by the ideal total processing time (IPT_i) as processing the shorter jobs first might be beneficial. There are thus four possible variations of ordering (all listed for the sake of completeness) *RLIPT-DD* (both $(n_i; IPT_i)$ in decreasing order), *RLIPT-II* (both $(n_i; IPT_i)$ in increasing order), *RLIPT-DI* (decreasing order of n_i then increasing order of IPT_i), and *RLIPT-ID* (increasing order of n_i then decreasing order of IPT_i).

In summary, we sort patients' planning requests in increasing (or decreasing) order of priority values defined formally in Table 3. The heuristics to examine are SPT, LPT, SIPT, LIPT, RLIPT-DD, RLIPT-II, RLIPT-DI, and RLIPT-ID.

Table 3. Priority definition.

Priority	Description	Formulation
PT	Total Processing Time	$\sum_{k \in \Theta_i} \sum_{o \in O} p_{i,k}^o$
IPT	Ideal Total Processing Time	$\sum_{k \in \Theta_i} IPT_{i,k}$
RLIPT	Request's Length and Ideal Total Processing Time	$(n_i; IPT_i)$

5.2. Genetic Algorithm

Meta-heuristics methods received more attention as its performance increased significantly to solve resource-constrained project scheduling for healthcare services [12,18,19,30]. In this problem, since it is straightforward to calculate the planning given a list of patients' entrance using FFS, we decided to use genetic algorithm to generate numerous lists and to converge toward the best.

The main elements of our GA are as follows:

- **Encoding:** We encode a chromosome as an ordinal sequence of patients, i.e., a chromosome is a permutation of the list of patients because we try to determine a priority list to schedule with FFS, similar to the priority-based heuristics earlier.
- **Initial population:** Given the permutation encoding scheme above, the initial population is created by randomly shuffling N_p times the set of patients $\{1, 2, \dots, n\}$.
- **Selection: Tournament Selector (TS)** is chosen because of its $O(n)$ time complexity and easy *parallelization* for speeding up calculation in practical use. Furthermore, while a large tournament results in loss of diversity, a binary tournament performance is generally inferior to a three-tournament. Hence, the tournament size is fixed by 3. Let us denote F_o ($0 \leq F_o \leq 1$) as the fraction of population selected to give offspring.

- **Crossover:** Due to the permutation encoding, the order of requests is critical to deduce the planning. Hence, we adopt **Partially Mapped Crossover** (PMX, Figure 5), first introduced by Grefenstette [31], so that a potentially good local order is transferred from parent to child.
- **Mutation:** The offspring are then applied **Partial Mutator Shuffle** (PSM, known as *Swap Mutator*, Figure 6) with probability P_M [32], so that a new order not in the current research neighborhood may appear.
- **Population management:** Denote $F_S (= 1 - F_O)$ as the fraction of current population that we want to keep for the next generation in order to preserve diversity. The population is kept constant by combining the children ($\lceil F_O \times N_P \rceil$) and parents ($\lfloor F_S \times N_P \rfloor$).
- **Stop condition:** The process is terminated after N_G successive generations that the objective value of the incumbent has not been changed.

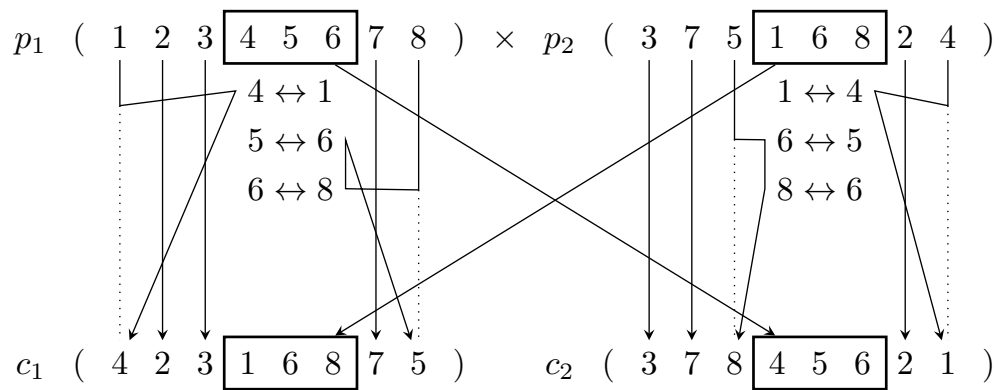


Figure 5. Example of PMX taken from [33]. Commencing with uniformly random two cut points, the subsequences passed to children ($p_1 \rightarrow c_2$; $p_2 \rightarrow c_1$) defines the element map to swap (continuously if necessary) in case of duplication when copying parent information to children.

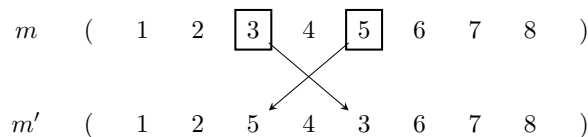


Figure 6. Example of PSM taken from [33]. Randomly choosing two distinct elements then swapping them.

We chose $(F_O, F_S) = (0.5, 0.5)$ for a balance of the intensity and the diversity of the search. The process stops after continuously staying at an objective value for long enough. The preliminary numerical experiments show that the incumbent rarely descends after 100 generations. Therefore, we can make the GA finish earlier by terminating it after a certain number of times, obtaining the same objective value instead of a fixed number of generations in total.

6. Computational Results

This section presents the benchmark based on historical data and then calibrates the proposed genetic algorithm. Finally, the results are shown and discussed.

6.1. Instance Generation

We use the test protocol and instances introduced in our previous work [2]. The dataset is published at [34]. Each instance has two sets of data: **resources availability** (seats M , doctors $D(j, h)$, nurses $I(j, h)$, pharmacy unit $F(j, h)$), and **planning requests** ($\{\Theta_1, \Theta_2, \dots, \Theta_n\}$ with n number of patients).

The generator's data is based on the data of Hospital Center of Troyes (CHT) in France (2020). The instances are generated as follows:

- The number of seats M is 30.
- Horizon is 4 weeks ($J = 28$), and each day consists of 22 time slots ($H = 22$).
- The maximal number of patients a nurse can monitor simultaneously: $\Psi = 6$.
- Human resources' presence are generated using a discrete uniform distribution (unif), of which the amount is an integer number varying between 0 (unavailable) and the maximum (fully available), noted as $\text{unif}[0; M]$ with M the major bound the resource capacity. Historical data gives us $N_D = 5$, $N_I = 6$ maximal number of doctors and nurses, respectively. Unlimited resource such as pharmacy could use 1 to represent to up state (open) and 0 for down state (close).

We divide further the resources' availability into four different scenarios to represent the reality:

- **Uniform:** $D(j, h) \sim \text{unif}[0; N_D]$; $I(j, h) \sim \text{unif}[0; N_I]$; $F(j, h) \sim \text{unif}[0; 1]$.
- **Daily:** Similar to **Uniform**, but the distribution is repeated *daily*.
- **Weekly:** Similar to **Uniform**, but the distribution is repeated *weekly*.
- **Weekend:** Similar to **Uniform** but no resource available on weekends, simulating the current working schedule at the service.
- For each request of indexed i patient, we define the following:
 - Number of required sessions $|\Theta_i| \sim \text{unif}[1; 3]$.
 - Time-lag $\Delta_{i,k} \sim \text{unif}[1; 3]$, except $\Delta_{i,1} = 0$.
 - Consultation time $p_{i,k}^A \sim \text{unif}[0; 1]$ (yes or no).
 - Installation time $p_{i,k}^B = 1$.
 - Drug mixing time $p_{i,k}^C \sim \text{unif}[0; 3]$.
 - Authorization of a priori drug mixing $e_{i,k}^C \sim \text{unif}[0; 1]$ (yes or no).
 - Treatment time $p_{i,k}^D \sim \text{unif}[0; 3]$.
- N_a , which is the total number of treatment sessions, takes a value from {15; 30; 60; 90; 120; 150; 180; 210}. In order to avoid service congestion, given the limited resources, we have fixed the maximum number of treatment sessions to 210.

Using the generator implemented by Java 8 with its native random engine, we created ten instances for each scenario each N_a , resulting in 320 test instances in total.

All of the numerical tests were executed on the same machine with CPU Intel Core i7-9750H (2.6 GHz~4.5 GHz), and the time limit and the specific implementation of each method are defined as follows:

- Solver CPLEX 12.10 solves the MILP within a time limit of 1 hour by using one CPU thread.
- Heuristic approaches are implemented purely on Java 8.
- Genetic algorithm's implementation based on Jenetic library 5.0 [35].

6.2. Calibration on the Genetic Algorithm

For each generated instance, we assign it to a difficulty level. The estimation of the difficulty level was based on the GAP found by MIP resolution with CPLEX after one hour of resolution. Three test instances were selected to represent the different levels of difficulty:

- **Easy** (low GAP): 90 weekly 3 ($GAP = 0.10\%$).
- **Normal** (medium GAP): 180 daily 8 ($GAP = 12.74\%$).
- **Hard** (high GAP): 210 daily 9 ($GAP = 44.40\%$).

First, $N_P = 1000$ was chosen for a large enough population. We observed the incumbent after 500 generations before picking a suitable number of steady generation N_G .

Furthermore, because P_M is usually small and P_C is frequently high, we tried the test with $P_M \in \{0.01, 0.1\}$ and $P_C \in \{0.8, 1.0\}$. For each instance and tuple $(P_C; P_M)$, we ran Algorithm 1 30 times and recorded the best objective value in order to trace the convergence.

In general, as we can see in Figure 7, the genetic algorithm generally converges after 100 generations. Therefore, we chose $N_G = 100$.

Algorithm 1: Pseudo-code for the genetic algorithm

Data: The set of planning requests $\Theta = \{\Theta_1, \Theta_2, \dots, \Theta_n\}$ where n is the number of input patients.

Result: An ordered permutation $\Theta^* = \{\Theta_{[1]}, \Theta_{[2]}, \dots, \Theta_{[n]}\}$ of requests to apply First Fit strategy.

begin

Initialize the population Π by inserting the randomly shuffled Θ until $|\Pi| = N_P$.

$current_best_chromosome \leftarrow null$.

$current_best_objective \leftarrow +\infty$.

$steady_counter \leftarrow 0$.

while $steady_counter \leq N_G$ **do**

Evaluate fitness function of individuals of Π .

Use selector **TS** to get $\lceil F_O \times N_P \rceil$ individuals from Π to mate by cross-over operator **PMX** with probability P_C .

Mutate the offspring by mutator **PSM** with probability P_M .

Use **TS** to get $\lfloor F_S \times N_P \rfloor$ survival individuals from Π for next generation.

Replace old generation: $\Pi \leftarrow survivors \cup offsprings$.

if $current_best_objective > objective\ value\ of\ the\ best\ chromosome\ of\ \Pi$ **then**

Save the best chromosome of Π to $current_best_chromosome$.

$current_best_objective \leftarrow objective\ value\ of\ current_best_chromosome$.

$steady_counter \leftarrow 0$.

else

$steady_counter \leftarrow steady_counter + 1$.

end

end

Return $current_best_chromosome$.

end

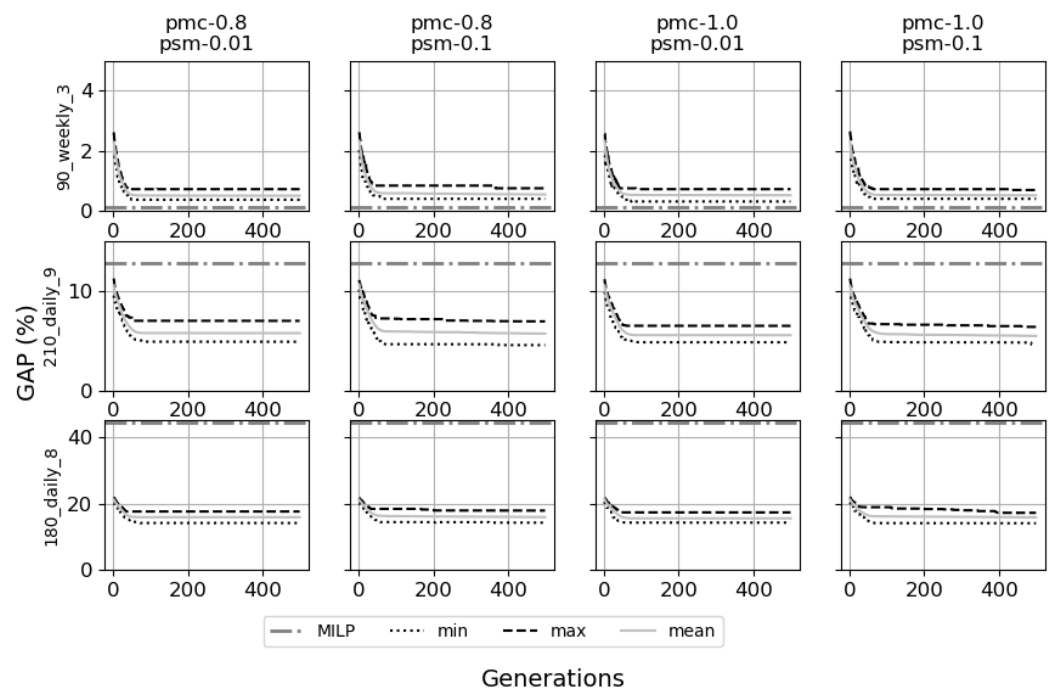


Figure 7. Evolution of GAP (%) for different difficulty and different values ($P_C; P_M$).

Furthermore, the GAP values of multiple runs on the same parameters stay close together, as we can see in Figure 7, and the difference between the best (min GAP) and the worst (max GAP) found objective values is always under 5% after 100 generations. Thus, the GA is converged (i.e., stable) with the corresponding parameters.

We observe a general decrease of final GAP when the mutator probability P_M going from 0.01 to 0.1 and the cross-over probability P_C increasing from 0.8 to 1.0 (Table 4 and Figure 7). Hence, the choice $(P_C; P_M) = (1.0; 0.1)$ is reasonable.

Table 4. Attained GAP (%) at the 500th generation for different difficulties and different values $(P_C; P_M)$. The best results are in bold.

Instances	P_C	P_M	Min	Max	Mean
180 daily 8	0.80	0.01	14.18	17.64	15.93
		0.10	14.28	17.92	15.96
	1.00	0.01	14.30	17.37	15.59
		0.10	14.12	17.24	15.88
210 daily 9	0.80	0.01	4.86	6.97	5.74
		0.10	4.54	6.92	5.69
	1.00	0.01	4.82	6.47	5.54
		0.10	4.60	6.36	5.45
90 weekly 3	0.80	0.01	0.37	0.71	0.50
		0.10	0.39	0.74	0.54
	1.00	0.01	0.31	0.72	0.52
		0.10	0.39	0.69	0.51

6.3. Numerical Test Results

The relative gap for method m on an instance is calculated using the following formula:

$$GAP_m = \left(\frac{\text{objective value found by } m}{\text{Optimal objective value or best LB found by MILP}} - 1 \right) \times 100\%$$

Figure 8 and Table 5 show similar behaviors for the different heuristic methods with small instances ($N_a < 90$). However, when the total number of treatment sessions grows ($N_a \geq 90$), CPLEX cannot find an optimal solution within the limit of 1 hour due to exponentially increased combinations (Figure 9). Additionally, from this point, the performance of solving methods can be grouped into three distinct tiers:

- **Low performance:** including *RLIPT-ID*, *RLIPT-II*, *SIPT*, *SPT* in which the GAP vastly climbed from under 10% to over 20%, even 35% in weekend’s scenario.
- **Medium performance:** including *RLIPT-DD*, *RLIPT-DI*, *LIPT*, *LPT*, in which the GAP increased arguably slower than **low** group and stayed overall below 15%.
- **Good performance:** including MILP and GA, which in most cases give lower GAPs than the **medium** group, generally under 10%.

In general, CPLEX is effective for solving the proposed MILP. Nevertheless, when the problem grows in size and resource availability congestion appears, the GA method yields better performance. Furthermore, GA is better in terms of execution time (Figure 9 and Table 6).

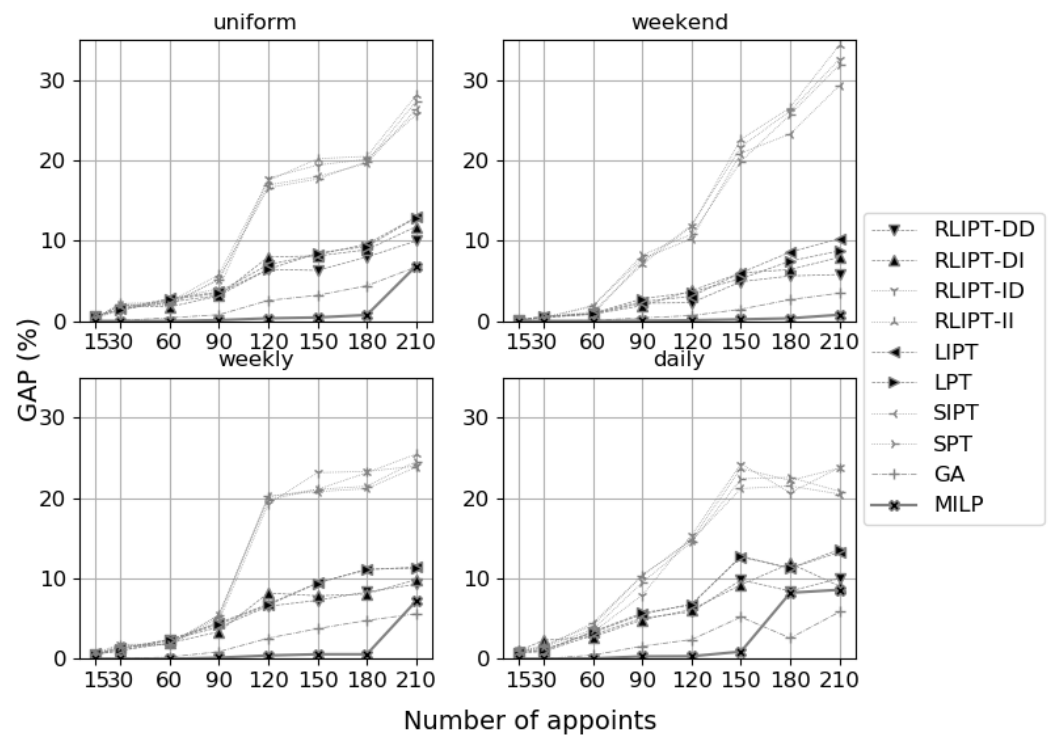


Figure 8. The relative gap of different approaches for varying instances sizes.

Table 5. Efficacy of different approaches (%). The best results are in bold.

Scenario	N_a	RLIPT-DD	RLIPT-DI	RLIPT-ID	RLIPT-II	LIPT	LPT	SIPT	SPT	GA	MILP
daily	15	0.75	0.96	1.22	0.95	0.92	0.75	1.01	0.84	0.00	0.00
	30	0.89	2.30	0.92	1.65	1.07	1.07	1.45	1.45	0.00	0.00
	60	2.95	2.71	3.31	4.14	3.27	3.36	4.44	3.63	0.45	0.04
	90	5.05	4.84	7.91	10.37	5.53	5.69	10.34	9.50	1.56	0.29
	120	5.84	6.12	15.39	14.81	6.80	6.71	14.73	14.51	2.39	0.32
	150	9.80	9.19	24.25	23.66	12.66	12.76	21.17	22.38	5.19	0.89
	180	8.42	11.87	20.64	22.17	11.26	11.26	21.49	22.55	2.57	8.21
	210	9.99	9.01	23.79	23.71	13.26	13.61	20.48	20.82	5.84	8.58
uniform	15	0.59	0.37	0.41	0.35	0.59	0.54	0.35	0.35	0.02	0.00
	30	1.41	1.95	1.29	2.16	1.54	1.40	2.02	1.99	0.06	0.00
	60	2.36	1.74	2.73	2.34	2.79	2.64	2.13	1.99	0.40	0.01
	90	3.19	3.13	4.02	5.77	3.61	3.47	5.16	5.19	0.78	0.11
	120	6.43	7.96	17.75	17.44	7.01	6.43	16.91	16.57	2.55	0.34
	150	6.29	8.09	19.39	20.19	8.26	8.46	17.95	17.66	3.20	0.44
	180	7.96	8.87	20.12	20.47	9.56	9.28	19.60	19.83	4.39	0.75
	210	9.96	11.75	25.68	28.08	12.91	12.86	26.36	27.34	6.74	6.86
weekend	15	0.08	0.12	0.10	0.13	0.14	0.15	0.08	0.11	0.00	0.00
	30	0.49	0.41	0.57	0.45	0.54	0.57	0.40	0.40	0.00	0.00
	60	0.90	0.85	0.82	1.15	0.92	0.94	1.94	1.80	0.08	0.00
	90	2.22	1.91	7.41	7.05	2.42	2.77	7.72	8.21	0.38	0.04
	120	2.28	3.77	11.89	11.77	3.07	3.57	10.24	10.80	0.68	0.10
	150	4.89	5.98	21.66	22.61	6.00	5.27	20.87	19.86	1.46	0.21
	180	5.60	6.41	26.14	26.56	8.58	7.45	23.28	25.67	2.67	0.34
	210	5.78	7.94	32.60	34.35	10.25	8.73	29.31	31.88	3.46	0.78

Table 5. Cont.

Scenario	N_a	RLIPT-DD	RLIPT-DI	RLIPT-ID	RLIPT-II	LIPT	LPT	SIPT	SPT	GA	MILP
weekly	15	0.55	0.54	0.74	0.60	0.61	0.58	0.49	0.55	0.00	0.00
	30	0.99	1.03	1.78	1.69	1.27	1.26	1.48	1.51	0.04	0.00
	60	2.30	1.96	1.84	1.77	2.40	2.35	1.80	1.76	0.21	0.01
	90	3.93	3.36	4.68	5.46	4.45	4.30	5.10	5.48	0.87	0.13
	120	6.53	8.18	19.23	20.12	6.79	6.69	20.25	19.77	2.56	0.41
	150	7.27	7.83	23.14	21.06	9.44	9.40	20.79	21.06	3.75	0.56
	180	8.31	8.06	23.29	23.14	11.06	11.16	21.15	21.46	4.77	0.56
	210	9.27	9.85	23.96	25.45	11.41	11.22	23.84	24.56	5.60	7.19

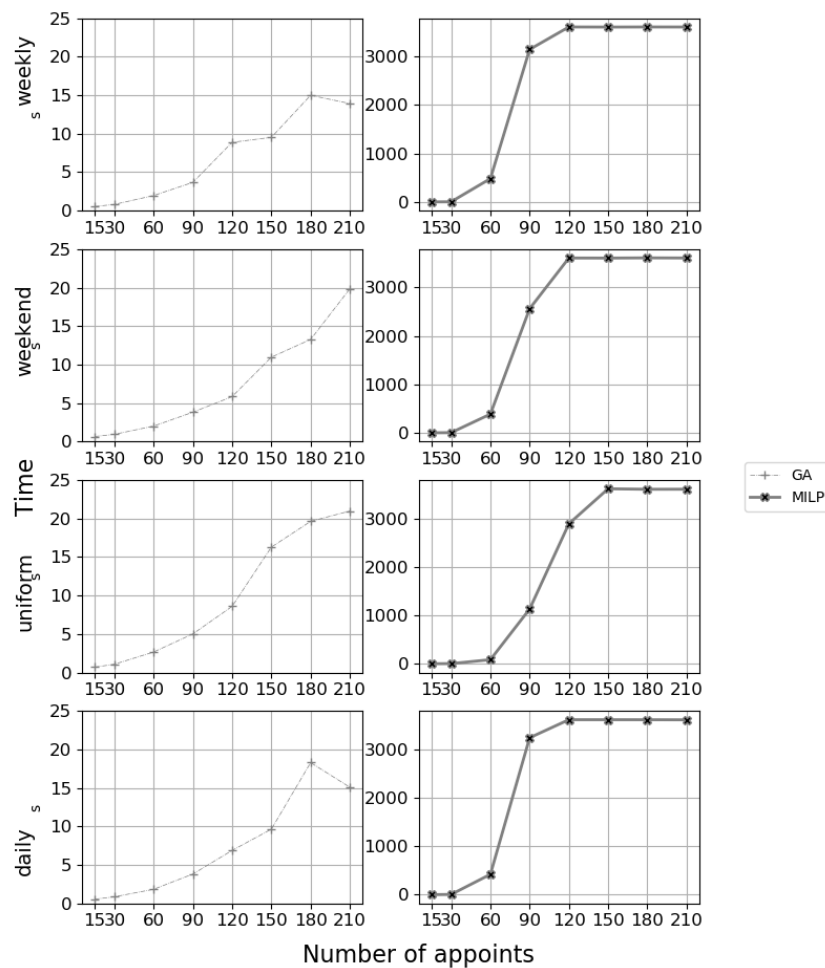


Figure 9. Execution time (in seconds) of different approaches. The priority-based heuristics give responses in several milliseconds, which is significantly faster than GA and MILP. We thus do not display them here.

Finally, regarding the heuristics, we can see their GAP is decided by how they prioritized the complicated jobs regarding the heuristics. The **low** group puts the *short* jobs at the top of their list, while the **medium** group plans the *long* jobs first. Hence, good heuristics should focus on the number of appointments and the total processing time.

On the other hand, almost no difference is observed between *RLIPT-DD* and *RLIPT-DI*: the priority lists of both methods put the lengthier jobs' sequences higher, but the former gives priority to longer processing times. At the same time, the latter arranges for a shorter processing time. In other words, tie-breaking by processing time has little effect. **The number of treatment sessions in a sequence** is probably critical for better performance, confirming our intuition for the RL IPT priority rule.

Furthermore, considering the ideal case of process, which helps reduce the weight of the processing time of *the drug mixing* stage, does not improve the results significantly, as seen in (*LIPT; LPT*), (*SIPT; SPT*).

Thus, *MTS (Most Total Successor)*, which is the incentive for the *RLIPT* method, has the best promising performance among the studied heuristics.

Table 6. Execution time (in seconds) of different approaches. The priority-based heuristics give responses in several milliseconds, which is significantly faster than GA and MILP. We thus do not display them here. The best results are in bold.

Scenario	N_a	GA	MILP
daily	15	0.54	2.64
	30	0.81	5.17
	60	1.96	481.84
	90	3.67	3144.82
	120	8.88	3607.37
	150	9.52	3605.55
	180	14.99	3607.27
	210	13.91	3606.57
uniform	15	0.65	3.15
	30	0.94	6.93
	60	2.01	388.04
	90	3.81	2560.58
	120	5.84	3605.26
	150	10.96	3603.58
	180	13.26	3607.83
	210	19.85	3605.19
weekend	15	0.69	1.86
	30	1.07	4.17
	60	2.66	85.48
	90	5.01	1131.40
	120	8.59	2892.92
	150	16.28	3614.94
	180	19.63	3604.99
	210	20.97	3606.20
weekly	15	0.54	2.86
	30	0.85	6.71
	60	1.83	420.77
	90	3.81	3236.47
	120	6.88	3608.38
	150	9.64	3608.42
	180	18.29	3607.80
	210	15.09	3606.19

7. Conclusions and Future Works

This work proposed a novel mathematical model for planning with time-varying resource constraints and strict time-lags, applied in a real-world outpatient chemotherapy service. The model included all four stages of each care session (consultation, installation, monitoring, and drug mixing), thus providing the most mathematically detailed description of the therapy. The formulation can be reused for more complex constraints such as restricting resource access and unavailability of patients to maximize patients' satisfaction.

Using an empirical test with real-world data, we demonstrated the efficiency of heuristics and meta-heuristics under the objective function of minimizing total completion time. The *Most Total Successor (MTS)* rule, which prioritizes lengthier sequences of treatment

sessions, should bring about practical use in planning management. On the other hand, the genetic algorithm proves to be an excellent candidate to handle complicated constraints as a alternative for commercial solver. In practice, the heuristic can be useful for quick and simple planning tools, while the GA might be useful for long-term and sustainable planning systems.

Finally, we are moving to an online version of this problem to provide an efficient algorithm that responds to the unpredictable fluctuation of resources and demands while leveraging the system's maximal operation.

Author Contributions: The authors contributed equally to the different steps of this research. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset is available at <https://github.com/qnhant5010/CHT-I> (accessed on 1 November 2021). All the data have been presented in the manuscript. The codes developed for the current study are available from the corresponding author if requested.

Acknowledgments: This work was supported by the Centre Hospitalier de Troyes (France), particularly the outpatient chemotherapy service.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Hesaraki, A.F. Tactical & Operational Models for Scheduling Chemotherapy Appointments. Ph.D. Thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2019.
- Tran, Q.N.H.; Chehade, H.; Nguyen, N.Q.; Laplanche, D.; Sanchez, S.; Sadeghi, R.; Dugardin, F. Optimization of Chemotherapy Planning. In Proceedings of the International Conference on Optimization and Learning (OLA2020), Cádiz, Spain, 17–19 February 2020.
- Condotta, A.; Shakhlevich, N. Scheduling Patient Appointments via Multilevel Template: A Case Study in Chemotherapy. *Oper. Res. Health Care* **2014**, *3*, 129–144. [[CrossRef](#)]
- Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Kan, A.H.G.R. Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. In *Annals of Discrete Mathematics; Volume 5, Discrete Optimization II*; Hammer, P.L., Johnson, E.L., Korte, B.H., Eds.; Elsevier: Amsterdam, The Netherlands, 1979; pp. 287–326. [[CrossRef](#)]
- Blazewicz, J.; Lenstra, J.K.; Kan, A.H.G.R. Scheduling Subject to Resource Constraints: Classification and Complexity. *Discret. Appl. Math.* **1983**, *5*, 11–24. [[CrossRef](#)]
- Weglarz, J. *Project Scheduling: Recent Models, Algorithms and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
- Wang, B.; Huang, K.; Li, T. Permutation Flowshop Scheduling with Time Lag Constraints and Makespan Criterion. *Comput. Ind. Eng.* **2018**, *120*, 1–14. [[CrossRef](#)]
- Bianco, L.; Caramia, M. An Exact Algorithm to Minimize the Makespan in Project Scheduling with Scarce Resources and Generalized Precedence Relations. *Eur. J. Oper. Res.* **2012**, *219*, 73–85. [[CrossRef](#)]
- Yalaoui, F.; Chu, C. Parallel Machine Scheduling to Minimize Total Completion Time with Release Dates Constraints. *IFAC Proc. Vol.* **2000**, *33*, 715–722. [[CrossRef](#)]
- Lenstra, J.K. Sequencing by enumerative methods. Ph.D. Thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, 1976.
- Nguyen, N.Q.; Yalaoui, F.; Amodeo, L.; Chehade, H.; Toggenburger, P. Electrical Vehicle Charging Coordination Algorithms Framework. In *Studies in Systems, Decision and Control*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 357–373. [[CrossRef](#)]
- Ernst, A.; Jiang, H.; Krishnamoorthy, M.; Sier, D. Staff Scheduling and Rostering: A Review of Applications, Methods and Models. *Eur. J. Oper. Res.* **2004**, *153*, 3–27. [[CrossRef](#)]
- Burke, E.K.; De Causmaecker, P.; Berghe, G.V.; Van Landeghem, H. The State of the Art of Nurse Rostering. *J. Sched.* **2004**, *7*, 441–499. [[CrossRef](#)]
- Gupta, D.; Denton, B. Appointment Scheduling in Health Care: Challenges and Opportunities. *IIE Trans.* **2008**, *40*, 800–819. [[CrossRef](#)]
- De Causmaecker, P.; Vanden Berghe, G. A Categorisation of Nurse Rostering Problems. *J. Sched.* **2011**, *14*, 3–16. [[CrossRef](#)]
- Berg, B.; Denton, B.T. Appointment Planning and Scheduling in Outpatient Procedure Centers. In *Handbook of Healthcare System Scheduling*; Hall, R., Ed.; Springer: Boston, MA, USA, 2012; Volume 168, pp. 131–154. [[CrossRef](#)]
- Saville, C.E.; Smith, H.K.; Bijak, K. Operational Research Techniques Applied throughout Cancer Care Services: A Review. *Health Syst.* **2017**, *8*, 52–73. [[CrossRef](#)] [[PubMed](#)]

18. Ahmadi-Javid, A.; Jalali, Z.; Klassen, K.J. Outpatient Appointment Systems in Healthcare: A Review of Optimization Studies. *Eur. J. Oper. Res.* **2017**, *258*, 3–34. [[CrossRef](#)]
19. Marynissen, J.; Demeulemeester, E. Literature Review on Multi-Appointment Scheduling Problems in Hospitals. *Eur. J. Oper. Res.* **2018**, *272*, 407–419. [[CrossRef](#)]
20. Artigues, C.; Gendreau, M.; Rousseau, L.M.; Vergnaud, A. Solving an Integrated Employee Timetabling and Job-Shop Scheduling Problem via Hybrid Branch-and-Bound. *Comput. Oper. Res.* **2009**, *36*, 2330–2340. [[CrossRef](#)]
21. Hahn-Goldberg, S.; Beck, J.C.; Carter, M.W.; Trudeau, D.M.; Sousa, P.; Beattie, K. Solving the Chemotherapy Outpatient Scheduling Problem with Constraint Programming. *J. Appl. Oper. Res.* **2014**, *6*, 10.
22. Liang, B.; Turkcan, A. Acuity-Based Nurse Assignment and Patient Scheduling in Oncology Clinics. *Health Care Manag. Sci.* **2016**, *19*, 207–226. [[CrossRef](#)] [[PubMed](#)]
23. Legrain, A.; Omer, J.; Rosat, S. A Rotation-Based Branch-and-Price Approach for the Nurse Scheduling Problem. *Math. Program. Comput.* **2020**, *12*, 417–450. [[CrossRef](#)]
24. Menting, J.A.; Menting, J. Planning and Scheduling under Uncertainty in Oncology Day Care. Master Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2014.
25. Turkcan, A.; Zeng, B.; Lawley, M. Chemotherapy Operations Planning and Scheduling. *IIE Trans. Healthc. Syst. Eng.* **2012**, *2*, 31–49. [[CrossRef](#)]
26. Nguyen, N.Q.; Yalaoui, F.; Amodeo, L.; Chehade, H.; Toggenburger, P. Solving a Malleable Jobs Scheduling Problem to Minimize Total Weighted Completion Times by Mixed Integer Linear Programming Models. In *Intelligent Information and Database Systems; Lecture Notes in Computer Science*; Nguyen, N.T., Trawiński, B., Fujita, H., Hong, T.P., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 286–295. [[CrossRef](#)]
27. Kolisch, R.; Hartmann, S. Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. In *Project Scheduling: Recent Models, Algorithms and Applications; International Series in Operations Research & Management Science*; Weglarz, J., Ed.; Springer: Boston, MA, USA, 1999; pp. 147–178. [[CrossRef](#)]
28. Swamidass, P.M. Shortest Processing Time (SPT). In *Encyclopedia of Production and Manufacturing Management*; Springer: Boston, MA, USA, 2000; pp. 686–687.
29. Alvarez-Valdes, R.; Tamarit, J.M. Heuristic Algorithms for Resource Constrained Project Scheduling: A Review and an Empirical Analysis. In *Advances in Project Scheduling*; Elsevier: Amsterdam, The Netherlands, 1989; pp. 113–134.
30. Leeftink, A.G.; Bikker, I.A.; Vliegen, I.M.H.; Boucherie, R.J. Multi-Disciplinary Planning in Health Care: A Review. *Health Syst.* **2018**, *9*, 95–118. [[CrossRef](#)] [[PubMed](#)]
31. Grefenstette, J.J. Genetic Algorithms for the Traveling Salesman Problem. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, Pittsburg, PA, USA, 24–26 July 1985; Psychology Press: Hove, UK, 1985.
32. Abdoun, O.; Abouchabaka, J.; Tajani, C. *Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem*; 2012; p. 18.
33. Larrañaga, P.; Kuijpers, C.; Murga, R.; Inza, I.; Dizdarevic, S. Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. *Artif. Intell. Rev.* **1999**, *13*, 129–170. [[CrossRef](#)]
34. Tran, Q.N.H. CHT-I. 2021. Available online: <https://github.com/qnhant5010/CHT-I> (accessed on 16 November 2021).
35. Wilhelmstötter, F. Jenetics: Java Genetic Algorithm Library. 2019. Available online: <https://jenetics.io/> (accessed on 21 March 2020).