

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A column generation-based heuristic for a rehabilitation patient scheduling and routing problem

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Xiao L., Zhen L., Laporte G., Baldacci R., Wang C. (2022). A column generation-based heuristic for a rehabilitation patient scheduling and routing problem. *COMPUTERS & OPERATIONS RESEARCH*, 148, 1-10 [10.1016/j.cor.2022.105970].

Availability:

This version is available at: <https://hdl.handle.net/11585/897412> since: 2023-06-09

Published:

DOI: <http://doi.org/10.1016/j.cor.2022.105970>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

Xiao, Liyang, Lu Zhen, Gilbert Laporte, Roberto Baldacci, and Chenghao Wang. "A Column Generation-Based Heuristic for a Rehabilitation Patient Scheduling and Routing Problem." *Computers & Operations Research* 148 (December 2022): 105970. <https://doi.org/10.1016/j.cor.2022.105970>.

The final published version is available online at: <https://doi.org/10.1016/j.cor.2022.105970>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)

When citing, please refer to the published version.

Highlights

- We introduce the rehabilitation scheduling and routing problem with real-life requirements
- We formulate the problem as an integer linear program
- We propose a column generation approach on a set partitioning based reformulation
- We use instances based on real data to test the performance of our proposed solution methods

A column generation-based heuristic for a rehabilitation patient scheduling and routing problem

Liyang Xiao^a, Lu Zhen^a, Gilbert Laporte^b, Roberto Baldacci^c, Chenghao Wang^{d,*}

^a*School of Management, Shanghai University, Shanghai, China*

^b*HEC Montréal, Québec, Canada*

^c*Department of Electrical, Electronic, and Information Engineering “Guglielmo Marconi”, University of Bologna, Cesena, Italy*

^d*CNRS, UMR 7253 Heudiasyc, Sorbonne Universités, Université de technologie de Compiègne, Compiègne, France*

Abstract

Rehabilitation is an important branch of the modern healthcare system. Every day, rehabilitation patients move in hospital campus to receive treatments from therapists. The long timespan of these treatment routes leads to several patients’ complaints and results in negative effects. However, scheduling the treatment routes for patients is a complex task for hospital managers. This study investigates a rehabilitation patient scheduling and routing problem, which focuses on reducing the timespan of patients’ treatment routes. This real-life motivated problem can be described as a combination of several interrelated traveling salesman problems with time windows (TSPTWs) and is difficult to solve. We formulate the problem as an integer linear program (ILP) and we develop a greedy heuristic called “route-first, schedule-second”. Then a column generation solution method is proposed on a set partitioning based reformulation of the original model. Specifically, a tailored genetic algorithm and several effective accelerating strategies are developed within the column generation method. Numerical experiments are conducted on 30 instances devised from real data to validate the efficiency of the proposed solution approaches. Experimental results show that our methodology can generate high-quality solutions efficiently and is therefore suitable to be applied in practice.

Keywords: Rehabilitation patient scheduling; Traveling salesman problem; Time windows; Integer linear program; Column generation

1. Introduction

Rehabilitation medicine is concerned with maintaining and recovering the health of people, often with serious disabilities due to physical or psychological defects. In many countries, rehabilitation medicine is an important part in modern healthcare system. The demand of rehabilitation service will continue to increase with the growth of the ageing population and the rise of chronic diseases

*Corresponding author

Email addresses: xiaoliyang0509@gmail.com (Liyang Xiao), lzheng@shu.edu.cn (Lu Zhen), gilbert.laporte@hec.ca (Gilbert Laporte), r.baldacci@unibo.it (Roberto Baldacci), chenghao.wang@hds.utc.fr (Chenghao Wang)

(Schimmelpfeng et al., 2012). In China for example, rehabilitation medicine is developing rapidly as there is a massive demand of rehabilitation treatments. According to the market analysis published by PricewaterhouseCoopers (PwC) in October 2016, it is conservatively estimated that over 100 million people are in need of rehabilitation services in China.

The patients who need rehabilitation services usually suffer from serious pain, consciousness disorders, moving and communication difficulty due to all sorts of medical issues such as illness (stroke, cerebral palsy), accidental injuries, surgical operations. Through rehabilitation treatment, these patients can achieve recovery from dysfunction, lower clinical disability rates, increase self-care ability and improved life quality. For example, a Copenhagen study (Jørgensen et al., 1995) showed that after receiving rehabilitation treatment, 80% of stroke patients who were initially unable to walk regained ambulation ability within six weeks, and 95% reached it within 11 weeks.

Rehabilitation patients often stay in rehabilitation hospitals for several weeks during the period of recovery, and every day they receive multiple rehabilitation treatments based on their own situation. A rehabilitation program is designed for each patient and includes various treatment items like functional training, massage, acupuncture, auditory and speech training, brain circuit training, etc. Table 1 gives examples of rehabilitation items. The patients’ treatments are provided by specific therapists according to doctor’s advice. In addition, the duration of each treatment is also pre-assigned. The therapists are affiliated with diverse hospital departments. Every day, patients need to go through these departments in order to get their rehabilitation treatments completed.

In most of rehabilitation hospitals, the treatment schedules are manually determined. The primary consideration of hospitals is to use the therapist resource in the best possible way. Scheduling these patients’ activities is highly complex, time consuming and very cumbersome, and the hospital manager usually follows a “first come is first reserved” rule. Due to the limitation of medical devices and therapist resources, the waiting time between successive treatments can be long. This leads to several complaints related to patients’ discomfort and inconvenience to their accompanying persons (nursing assistants or relatives). Moreover, it indeed diminishes patient compliance.

Table 1: Examples of rehabilitation items

Department	Rehabilitation item	
Instrument	Brain circulation	Intermediate frequency
	Laser	Low frequency
	Lymphatic drainage	Millimeter wave
	Transcranial magnetic stimulation	Ultrasound
Physical treatment	Functional training	Low limbs rehabilitative robot
Traditional rehabilitation	Acupuncture	Massage
Training	Swallowing	Speech and language

Motivated by a real-life application, this study aims to address the rehabilitation patient scheduling problem which is called the rehabilitation patient scheduling and routing problem (RP-SRP), which contains features of the traveling salesman problem with time windows (TSPTW) and

of some machine scheduling problems. The RPSRP is a new variant of TSPTW. To be specific, RPSRP can be seen as a combination of a large number of interrelated TSPTWs, each TSPTW problem copes with a dedicated route for each patient, and has a complicated structure.

The contributions of this paper are as follows: (1) we propose a new RPSRP model on the basis of a comprehensive consideration of real factors involved during patient’s rehabilitation activities; (2) we reformulate the original RPSRP model as a set partitioning model, which contains a master problem and a number of pricing subproblems; (3) we propose two approximate approaches, based on a greedy scheme and on column generation techniques, and we solve the RPSRP model and the set partitioning based model within a reasonable computation time.

The remainder of this paper is organized as follows. Section 2 reviews the literature. Section 3 describes the problem and presents a compact mathematical model of the RPSRP. Section 4 introduces a greedy heuristic, named “route-first, schedule-second” (RFSS). Section 5 describes a set partitioning based model and presents a column generation based approach. Section 6 conducts experiments to validate the proposed approaches. Section 7 concludes the study and provides perspectives.

2. Related works

Operations research (OR) in health care has received significant attention for almost three decades. Countless efforts have been made to increase hospital cost control strategies and operational efficiencies. As a result, numerous studies exist on scheduling various activities for patients. Detailed literature reviews are provided in Ahmadi-Javid et al. (2017), Cayirli and Veral (2003) and Gupta and Denton (2008) for appointment scheduling, and in Samudra et al. (2016) and Hamid et al. (2019) for operating room scheduling. Concerning these topics, the goals and scope have been limited to resources within a single clinical department or discipline. However, due to the multi-discipline nature of rehabilitation care, resources from multiple departments are involved when considering the scheduling of rehabilitation patients. The proposed RPSRP is analogous to the rehabilitation scheduling problem. Additionally, the basic problem in RPSRP is closely related to the home health care scheduling and routing problem (HHCSRP) (Grenouilleau et al., 2019). In this section, we mainly survey the literature relative to these two relevant topics.

The rehabilitation scheduling problem is identified as an application variant of multi-appointment scheduling problems in hospital (MASPHs) (Marynissen and Demeulemeester, 2019). Despite the importance of rehabilitation medicine, only a few papers have dealt with rehabilitation scheduling problems. In particular, Chien et al. (2008) aimed at reducing the patient’s waiting time during the process of their rehabilitation treatments, and proposed a genetic algorithm (GA) for the problem. Later, different approaches have been applied to tackle the rehabilitation scheduling problem: GA with data mining technique (Chien et al., 2009), hybrid GA with 2D encoding (Huynh et al., 2018), and improved cuckoo search (Xiao et al., 2018). Also, Zhao et al. (2015) proposed a bi-objective GA to balance patient’s waiting time against the makespan. These studies formulated the rehabilitation scheduling problem as a hybrid shop scheduling problem (HSSP). Patients are seen as

products and therapists are regarded as machines with the same timeslot. In practice, treatment start time usually is restricted by specific time windows associated with therapists. The therapists are geographically scattered over the hospital campus and thus a travel time is incurred when a patient moves between two therapists.

As another active research field of OR in health care, HHCSRP has drawn considerable attention in recent years, it involves dispatching at least cost caregivers or therapists to serve patients in their homes. For a comprehensive overview on HHCSRP, the reader is referred to the excellent review work of Fikar and Hirsch (2017). There are similarities between HHCSRP and RPSRP: one is about their common health-care context; the other is about their common features, since both of them are related to routing problems such as the famous traveling salesman problem (TSP) and vehicle routing problem (VRP). In the HHCSRP, patients stay in their homes and caregivers or therapists are scheduled and routed to perform various services. However in the RPSRP, the operational mode is reverse: the therapists are stable and patients need to move. Furthermore, it is necessary to point out that in the HHCSRP, each node is allowed to be visited exactly once or at most once; yet in RPSRP, it is implied that each node (i.e., therapist) must be visited by multiple patients. This highlights the key dissimilarity between HHCSRP and the proposed RPSRP from the perspective of modeling. The model combines features of both routing problems and machine scheduling problems: when routing problems are solved for patients, the side constraints of machine scheduling problem need to be respected.

The daily rehabilitation activities of each patient can be viewed as a treatment route. A treatment route is a path in which a patient starts and ends at the ward and visits different therapists to get his or her rehabilitation program completed. Determining a treatment route means solving a TSP. Also, the therapists must be reached within specific time windows. That is to say, each patient has to solve for himself or herself a TSPTW. Since each therapist is visited multiple times by several patients, the treatment routes of the patients are interrelated. On the other hand, the therapists remain at the therapy rooms and treat the patients in sequence, i.e., each therapist can only treat one patient at a time. Each therapist can be viewed as a machine and each patient as a product; the problem can then be related to machine scheduling problems. In particular, if no precedence restriction is imposed on treatments, the problem becomes an open shop scheduling problem (OSSP). For comprehensive reviews of OSSP, we refer the readers to Ahmadian et al. (2021).

3. Problem description and mathematical formulation

In this section, we first formally describe the RPSRP, followed by the description of a compact mathematical formulation.

3.1. Problem description

The RPSRP can be described as follows. Let $G = (N \cup \{0\}, A)$ be a directed graph, $N = \{1, \dots, n\}$ is a set of nodes where the therapists are located and 0 is the patients' ward (henceforward

referred to as ward or depot). Each therapist node $i \in N$ has a specified time window $[w_i, \bar{w}_i]$, during which he or she is available to serve the patients. $A = \{(i, j) | i, j \in N \cup \{0\}\}$ is an arc set. Each arc $(i, j) \in A$ connects node i and node j and has a travel time t_{ij} . Consider a patient set $P = \{1, \dots, m\}$. Each patient $u \in P$ has to carry out a treatment route: departing from the ward; visiting a subset of known therapists denoted by N_u ($N_u \subseteq N$) thereby to receive rehabilitation treatments from them; and returning to the ward. Let o_{iu} be patient u 's treatment is assigned to therapist $i \in N_u$, and a treatment duration d_{iu} is associated with each o_{iu} . Also, a relax time r_{iu} is usually required by patients when a treatment is completed.

We make the following assumptions:

- The treatment route of each patient must start and end at the same ward;
- Each patient is pre-assigned a subset of therapists to be visited;
- Each therapist is visited at most once by the same patient;
- There is no precedence restriction on treatments;
- No overlap of time is allowed between each treatment of a patient by a therapist;
- A relax time is required by patients after receiving a treatment from therapist.

The RPSRP aims at determining a set of treatment routes with minimized timespan for all patients.

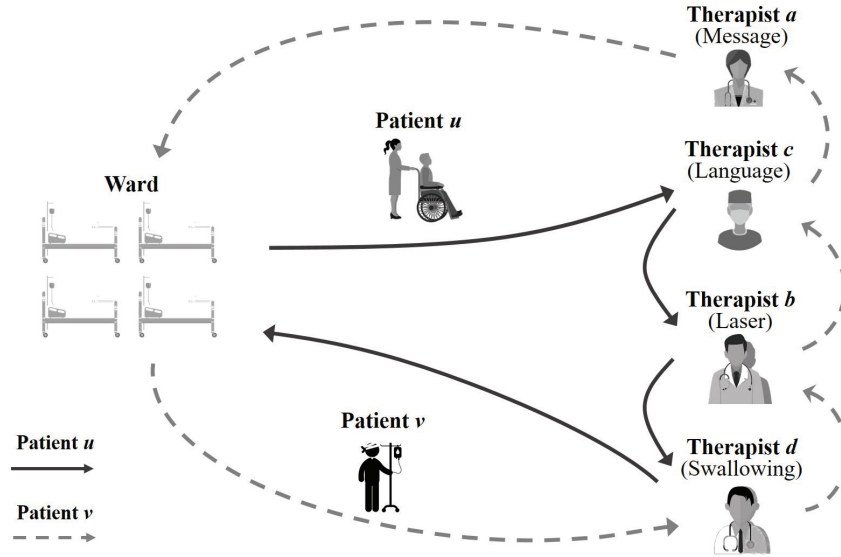


Figure 1: Illustrative example of the RPSRP

Consider two patients u and v in need of rehabilitation care. To get their rehabilitation programs completed, patient u is allocated to visit therapists b, c, d , and patient v to therapists a, b, c, d . Figure 1 shows the possible treatment routes for these two patients, and the sequence of events contained in patient u 's treatment route is illustrated in Figure 2 .

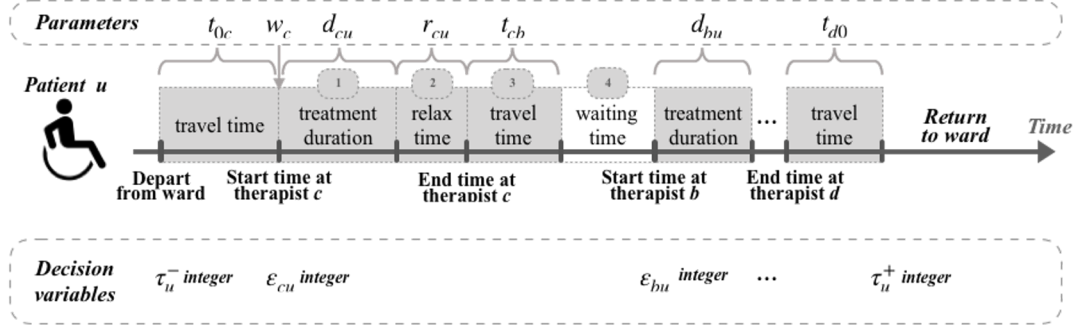


Figure 2: Events and time points of the treatment route of patient u

3.2. Objective of the RPSRP

The timespan of a treatment route is the total time elapsed from leaving depot to returning to the depot, which consists of the total travel time, the duration of treatments and the total waiting time. For each patient u , the duration of treatments is a constant, computed as $\sum_{i \in N_u} (d_{iu} + r_{iu})$. Therefore the goal of RPSRP is to simultaneously optimize the patients' total travel and waiting time.

(1) *Travel time*: In the proposed version, although the allocation between patients and therapists is known in advance, the sequences that patients visit their own allocated therapists are not imposed: they are decision variables. Obviously, the total travel time is directly determined by the visiting sequences of patients' treatment routes.

(2) *Waiting time*: The waiting time of a patient refers to the time passed between the arrival time at a therapist and the start time of the treatment. For each treatment o_{iu} ($u \in P, i \in N_u$), a treatment start time ϵ_{iu} has to be planned. Patient u arriving at therapist i earlier than ϵ_{iu} is allowed but induces a waiting time (as illustrated by phase ④ in Figure 2). Reducing the waiting time of patients has its practical meaning. In rehabilitation hospitals, patients have to undergo multiple treatments, the waiting time of patients not only increases the timespan of their treatment routes, but also significantly influences the efficiency of hospitals' operations.

3.3. Mathematical formulation

Sets and indices

i, j : index of therapist node and depot;

u, v : index of patient;

t : index of time step;

N_u : subset of therapist nodes of patient u ;

N'_u : subset of therapist nodes of patient u plus the depot, i.e., $N'_u = N_u \cup \{0\}$;

P : set of patients;

P_i : subset of patients who need to visit therapist i ;

T : set of time steps of therapists;

Parameters

- $[w_i, \bar{w}_i]$: time window of therapist i ;
- t_{ij} : travel time from therapist i to therapist j ;
- d_{iu} : treatment duration of treatment o_{iu} (i.e., patient u 's treatment at therapist i);
- r_{iu} : relax time for patient u in ending the treatment o_{iu} ;

Variables

- x_{iju} : binary, equals one if patient u arrives at therapist j immediately after leaving therapist i , and zero otherwise;
- y_{uvi} : binary, equals one if therapist i treats patient u before patient v , and zero otherwise;
- ε_{iu} : integer, the start time of treatment o_{iu} ;
- τ_u^- : integer, start time of patient u 's treatment route;
- τ_u^+ : integer, end time of patient u 's treatment route.

The RPSRP can be formulated as a compact model, denoted by CM, as follows:

$$[\text{CM}] \quad \text{minimize} \quad \sum_{u \in P} (\tau_u^+ - \tau_u^-) \quad (1)$$

subject to

$$\sum_{i \in N'_u, i \neq j} x_{iju} = 1 \quad u \in P, \quad j \in N'_u \quad (2)$$

$$\sum_{j \in N'_u, i \neq j} x_{iju} = 1 \quad u \in P, \quad i \in N'_u \quad (3)$$

$$w_i \leq \varepsilon_{iu} \quad u \in P, \quad i \in N_u \quad (4)$$

$$\varepsilon_{iu} + d_{iu} \leq \bar{w}_i \quad u \in P, \quad i \in N_u \quad (5)$$

$$\varepsilon_{iu} + d_{iu} + r_{iu} + x_{iju}t_{ij} \leq \varepsilon_{ju} + (1 - x_{iju})\bar{w}_i \quad u \in P, \quad i, j \in N_u, \quad i \neq j \quad (6)$$

$$(w_i - \bar{w}_i)(1 - y_{uvi}) + \varepsilon_{iu} + d_{iu} \leq \varepsilon_{iv} \quad i \in N, \quad u, v \in P_i, u \neq v \quad (7)$$

$$y_{uvi} + y_{vui} = 1 \quad i \in N, \quad u, v \in P_i, u \neq v \quad (8)$$

$$\tau_u^- \geq \varepsilon_{ju} - t_{0j} + (t_{0j} - \bar{w}_j)(1 - x_{0ju}) \quad u \in P, \quad j \in N_u \quad (9)$$

$$\tau_u^- \leq \varepsilon_{ju} - t_{0j} + (\bar{w}_j + t_{0j})(1 - x_{0ju}) \quad u \in P, \quad j \in N_u \quad (10)$$

$$\tau_u^+ \geq \varepsilon_{iu} + d_{iu} + r_{iu} + t_{i0} - (t_{i0} + \bar{w}_i)(1 - x_{i0u}) \quad u \in P, \quad i \in N_u \quad (11)$$

$$\tau_u^+ \leq \varepsilon_{iu} + d_{iu} + r_{iu} + t_{i0} + \bar{w}_i(1 - x_{i0u}) \quad u \in P, \quad i \in N_u \quad (12)$$

$$\tau_u^+ \geq \tau_u^- \quad u \in P \quad (13)$$

$$x_{iju} \in \{0, 1\} \quad u \in P, \quad i, j \in N'_u, \quad i \neq j \quad (14)$$

$$y_{uvi} \in \{0, 1\} \quad i \in N, \quad u, v \in P_i \quad u \neq v. \quad (15)$$

Objective function (1) minimizes the overall timespan of all patients' treatment routes. Constraints (2)–(7) describe the treatment route of each patient as a TSPTW. Constraints (2)–(3) impose that for each therapist included in a treatment route, it has both a predecessor and a successor. Constraints (4) and (5) ensure that the treatments are provided during corresponding therapists' time windows. In addition, constraints (6) define the correct treatment start times by guaranteeing the travel time between two consecutive treatments of a treatment route. If patient u is allocated to be treated by therapists i and j successively, (i.e., $x_{iju} = 1$), patient u 's treatment start time ε_{iu} plus the treatment duration d_{iu} , plus the relax time r_{iu} , and plus the travel time t_{ij} , must not exceed patient u 's treatment start time ε_{ju} . The gap between them is the possible waiting time of patient u at therapist j (as illustrated by phase ④ of Figure 2). Constraints (7) and (8) indicate that each therapist can only treat one patient at a time. When $y_{uvi} = 1$, therapist i treats patient u ahead of patient v , then $\varepsilon_{iu} + d_{iu} \leq \varepsilon_{iv}$. Constraints (9) and (10) compute the start time of each patient's treatment route. When $x_{0ju} = 1$, therapist j provides the first treatment of patient u , and hence $\tau_u^- = \varepsilon_{ju} - t_{0j}$. Similarly, constraints (11) and (12) determine the end time of each patient's treatment route, where $x_{i0u} = 1$ means that therapist i provides the last treatment of patient u . Finally, constraints (13)–(15) define feasible domains of the decision variables.

4. A greedy heuristic: “route-first, schedule-second” (RFSS)

The RPSRP minimizes the overall timespan of all patients' treatment routes, which contains two parts: the total travel time and the total waiting time. The former is depended on the visit sequence of patients' treatment routes while the latter is determined by the start times of the treatments. In this section, we propose a greedy heuristic, referred to as “route-first, schedule-second” (RFSS). The decision variables are fixed in two phases: (1) the “route” phase: determining visiting sequence of treatment route for each patient; (2) the “schedule” phase: scheduling the treatments for each therapist. In particular, the “route” phase calculates a lower bound on the optimal value of formulation CM.

The greedy heuristic RFSS is outlined in Algorithm 1. The “route” phase solves a relaxation of RPSRP, which decomposes the problem into a set of independent asymmetric TSP (ATSPs) (Glover et al., 2001), where the travel time between two nodes i, j is defined as $d_{iu} + r_{iu} + t_{ij}$. More specifically, for each patient $u \in P$, the first stage is to find the shortest path of treatment route with a minimum timespan, i.e., duration of travel time. Note that the constraints related to therapists (e.g., time windows, sequence to treat patients) are dealt with in the “schedule” phase. Once the shortest treatment routes of patients are obtained, the treatment sequence of each patient is also determined; and consequently the problem is reduced to a job shop scheduling problem (JSSP) with setup times, deadlines and precedence constraints (Balas et al., 2008), where the processing time is d_{iu} , setup time of a job on two consecutive machines i, j is defined as $r_{iu} + t_{ij}$, and the deadline

of each job associated with an operation is given by the available interval $[w_i, \bar{w}_i]$ of corresponding n machines (i.e., the therapists). Then, the “schedule” phase sets the start times of the treatments to obtain each patient’s waiting time.

Algorithm 1: A greedy heuristic: “route-first, schedule-second” (RFSS)

```

1 /* “route” phase */
2 for Each patient do
3   | Find the shortest path for treatment route with a minimum timespan;
4 end
5 /* “schedule” phase */
6 for Each treatment route do
7   | Calculate the earliest start time of each treatment  $\varepsilon_{iu}^*$  ;
8 end
9 Sort the treatments in chronological order of their earliest start times  $\varepsilon_{iu}^*$  ;
10 for Each treatment with earliest start time  $\varepsilon_{iu}^*$  sorted in chronological order do
11   | /*Assign revised start time  $\varepsilon'_{iu}$  to each treatment*/
12   | if Therapist  $i$  serves another treatment (say  $o_{iv}$ ) at time  $\varepsilon_{iu}^*$  then
13     | Incumbent treatment  $o_{iu}$  is served until therapist is available:  $\varepsilon'_{iu} = \varepsilon'_{iv} + d_{iv}$  ;
14     | Postpone  $\varepsilon_{iu}^*$  for all successive treatments of patient  $u$ ;
15     | Update the chronological order of unassigned treatments;
16   | end
17   | else
18     |  $\varepsilon'_{iu} = \varepsilon_{iu}^*$ 
19   | end
20 end

```

Both the ATSP and the JSSP are NP-hard in the strong sense, their combination makes the problem even more difficult. However in real-world applications of RPSRP, the number of therapists to be visited by each patient is actually small, e.g., at most 10 treatments during one day, and therefore the “route” phase can be solved efficiently by using an off-the-shelf solver. Meanwhile, the number of treatments is quite large. An efficient timetabling procedure is used in “schedule” phase to set start times of treatments. For each therapist $i \in N$, no treatment can be operated before w_i . The “schedule” phase firstly sets an earliest treatment start time ε_{iu}^* at each therapist that included in the route. However, the resources of therapists are constrained and overlaps of therapist times may occur. To eliminate the overlap of time on each therapist, the start times of treatments are revised in sequence, denoted as ε'_{iu} , by following the “first-come, first-served” (FCFS) rule. More specifically, if therapist i is available to serve treatment o_{iu} at ε_{iu}^* , then $\varepsilon'_{iu} = \varepsilon_{iu}^*$. Otherwise, if $\varepsilon'_{iv} < \varepsilon_{iu}^* < \varepsilon'_{iv} + d_{iv}$, patient v is served by therapist i before patient u , there exists an overlap between treatments o_{iv} and o_{iu} , hence the upcoming treatment o_{iu} is postponed until the termination of the incumbent treatment o_{iv} , i.e., $\varepsilon'_{iu} = \varepsilon'_{iv} + d_{iv}$ and the ε_{iu}^* of successive treatments of patient u are also postponed.

Figure 3 illustrates the solution to the small instance given by Tables 2 and 3. Given the routes of patients u and v (i.e., “0-c-b-d-0” and “0-d-b-c-a-0”), the values of ε_{iu}^* are given in Table 3. The

241 treatments corresponding to these two routes are o_{cu}, o_{bu}, o_{du} and $o_{dv}, o_{bv}, o_{cv}, o_{av}$, respectively.
 242 The earliest start time of each treatment, i.e., ε_{iu}^* , is computed assuming the first treatment of
 243 each patient is served at the earliest available time of the associated therapist. The treatments
 244 are then sorted in chronological order of their earliest start times, i.e., " $o_{cu}, o_{dv}, o_{bv}, o_{bu}, o_{cv}, o_{du},$
 245 o_{av} ". Under the chronological sequence, the treatments of patients u and v (i.e., o_{bu} and o_{bv}) have
 246 conflicts on therapist b 's timeslot. Treatment o_{bu} is therefore postponed until the termination of
 247 treatment o_{bv} , as shown in the upper part of Figure 3. The successor of treatment o_{bu} is o_{du} , which
 248 is also postponed together with o_{bu} . The new values ε'_{iu} are then computed, as shown in Table 3.

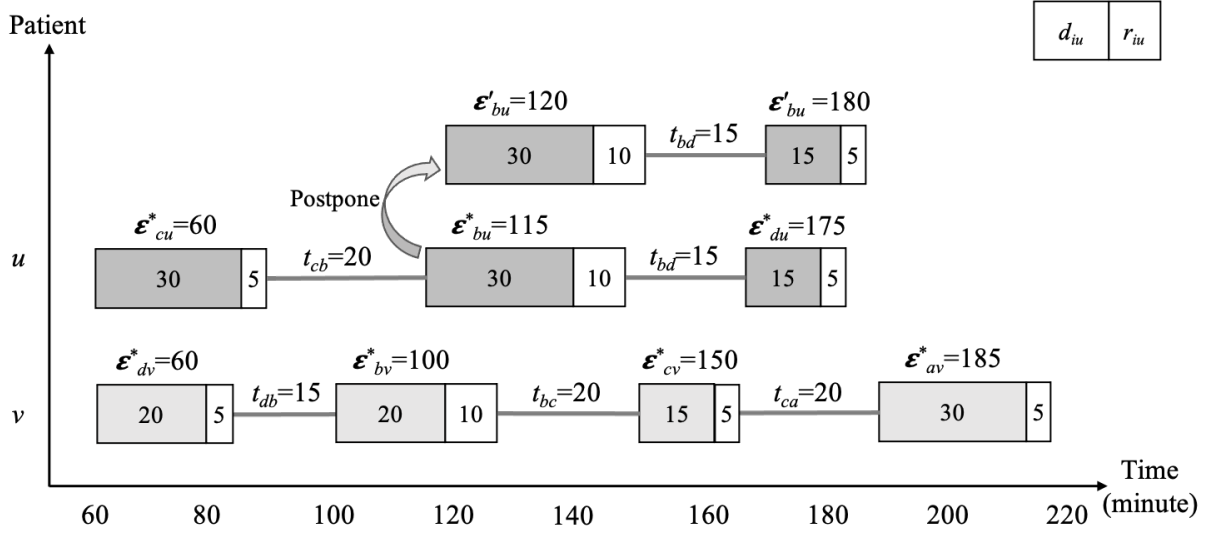


Figure 3: Illustration of the greedy assignment for a processing treatment

Table 2: Instance used to illustrate algorithm RFSS

Node	Time window	Arc	Traveling time
0	[0, 1000]	(0,a)	30
a	[70, 600]	(0,b)	35
b	[75, 605]	(0,c)	35
c	[60, 595]	(0,d)	25
d	[60, 600]	(a,b)	25
		(a,c)	15
		(a,d)	20
		(b,c)	20
		(b,d)	15
		(c,d)	40

Table 3: Treatment routes for the example of Table 2

Patient	Therapist	Treatment duration	Relax time	Start time	
				ε_{iu}^*	ε'_{iu}
u	b	30	10	60	60
	c	30	5	115	120
	d	15	5	170	175
v	a	30	5	185	185
	b	20	10	100	100
	c	15	5	150	150
	d	20	5	60	60

5. A column generation based approach

The RPSRP consists of several interrelated TSPTWs and involves features of the OSSP. Both the TSPTW and OSSP are NP-hard (Focacci et al., 2002; Gu  ret and Prins, 1999). Moreover, according to Savelsbergh (1985), even finding a feasible solution of the TSPTW is NP-complete. The aforementioned RFSS heuristic may generate a costly solution far from the optimal one. Therefore, in this section, we describe an effective heuristic column generation algorithm, also capable of solving large-scale instances of the RPSRP. The algorithm solves the LP-relaxation of a set partitioning based formulation of the RPSRP in a column generation fashion, heuristically solving the associated pricing problem by means of a genetic algorithm.

5.1. A set partitioning based mathematical formulation for the RPSRP

In this section, we propose a set partitioning based formulation, in which each variable represents a specific treatment route. Define the set of feasible treatment routes for patient u as D_u . Define the cost of patient's treatment route d , i.e., the timespan of route d as z_d . Therefore, for a given treatment route d of patient u , the cost of the route can be computed as $z_d = \tau_u^+ - \tau_u^-$. Each treatment route consumes two types of resources, therapists and time steps. Define binary parameters θ_{djt} such that $\theta_{djt} = 1$ if therapist j is assigned in treatment route $d \in D_u$ at time step t , and $\theta_{djt} = 0$ otherwise. Define binary variable x_d such that $x_d = 1$ if treatment route d is selected in the solution and $x_d = 0$ otherwise.

Given the above notations, we can formulate the problem as a set partitioning based model, called the master problem (MP) as follows.

$$[\text{MP}] \quad \text{minimize} \quad \sum_{u \in P} \sum_{d \in D_u} z_d x_d \quad (16)$$

subject to

$$\sum_{d \in D_u} x_d = 1 \quad u \in P \quad (17)$$

$$\sum_{u \in P} \sum_{d \in D_u} \theta_{djt} x_d \leq 1 \quad t \in T, j \in N \quad (18)$$

$$x_d \in \{0, 1\} \quad d \in D_u, u \in P \quad (19)$$

The objective (16) minimizes the total timespan of the selected treatment routes. Constraints (17) ensure that one treatment route is selected for each patient. Constraints (18) state that at most one patient is treated by a therapist at any time step.

The master problem (MP) is an integer linear programming (ILP) model, which contains $|P|+|T||N|$ constraints. However, due to the very large number of feasible treatment routes, it is impractical and intractable to enumerate all feasible treatment routes. By relaxing the integrality constraint (19), the linear relaxation of the MP (referred as LMP) can be solved via column generation, where a column represents a treatment route. In practice, there is no way to explicitly enumerate all columns and therefore the LMP can not be solved directly. Instead, only a small portion of columns are generated first and a restricted LMP (denoted as RLMP) is solved, and the dual value of each constraint of the RLMP is computed. Based on the dual values, the pricing subproblems are then solved to construct additional columns (i.e., the ones with negative reduced cost). These new columns are contributing to the improvement of the RLMP, if they are found, they are then added into the RLMP, and the updated RLMP is solved again. This iterative process is repeated until no new columns are found. At this point, the optimal solution of the current RLMP is also optimal for the LMP. For a general treatment of column generation techniques, the reader is referred to Lübbecke and Desrosiers (2005).

Algorithm 2 summarizes the main steps of the column generation procedure and the proposed solution algorithm. The algorithm first solves problem LMP in a column generation fashion. If the final solution is integer, a feasible RPSRP solution has been computed. Otherwise, a reduced problem MP obtained by substituting the set of treatment routes with the set of routes computed during the column generation phase is solved to optimality using a general purpose MIP solver. Below, we describe the pricing problem associated with formulation LMP, and we give some details about the different components of the algorithm. In particular, we describe a heuristic algorithm to solve the pricing problem.

5.2. Pricing problem

Let α_u and β_{tj} be the dual variables for constraints (17) and (18), respectively. Then the reduced cost of treatment route d performed by patient u is then computed as

$$z_d - \alpha_u - \sum_{j \in N} \sum_{t \in T} \beta_{tj} \theta_{djt} . \quad (20)$$

Since each patient has to follow one treatment route, each patient $u \in P$ corresponds to a subproblem. For the RPSRP with m patients, there are m subproblems in the pricing process. The parameters that are associated with a patient's treatment route d (i.e., θ_{djt}) in the master problem becomes decision variables for the subproblem. Subsequently, the decision values of the dual variables, i.e., α_u and β_{tj} obtained from master problem are passed as parameters to the subproblem. For a fixed patient $u \in P$, the pricing subproblem SP_u is to find a patient's treatment

Algorithm 2: Column generation based algorithm

```

1 Populate the initial RLMP (see Section 5.4.1);
2 while Any patient  $u \in P$  has a treatment route with a negative reduced cost and
   Total_Elapsed_Time  $\leq$  Total_Time_Lim do
3   Solve problem RLMP;
4   for Each patient  $u$  do
5     Generate a treatment route  $d \in D_u$  for  $SP_u$  (see Section 5.3);
6     if Treatment route  $d$  has a negative reduced cost then
7       | Update the RLMP;
8     end
9   end
10 end
11 if An integer MP solution has been found then
12   | return the solution found;
13 end
14 else
15   | Solve the reduced MP problem defined by the set of treatment routes forming RLMP;
16 end

```

route with the most negative reduced cost, and can be mathematically formulated as follows:

$$[SP_u] \quad \text{minimize } z_d - \alpha_u - \sum_{j \in N} \sum_{t \in T} \beta_{tj} \theta_{djt} \quad (21)$$

subject to

308

309 (2)–(6), (9)–(14) and

$$\sum_{t \in T} \theta_{djt} = d_{ju} \quad j \in N_u \quad (22)$$

310

$$\varepsilon_{ju} \leq t - (\theta_{djt} - 1)|T| \quad j \in N_u, t \in T \quad (23)$$

311

$$t - (1 - \theta_{djt})|T| \leq \varepsilon_{ju} + d_{ju} - 1 \quad j \in N_u, t \in T \quad (24)$$

312

$$\theta_{djt} \in \{0, 1\} \quad j \in N_u, t \in T. \quad (25)$$

313 The objective function (21) minimizes the reduced cost of the treatment route. The constraints
 314 of the pricing subproblem are complemented by constraints (22)–(25). Constraints (22) ensure that
 315 certain time steps of each relevant therapist (i.e., $j \in N_u$) should be reserved for the treatment
 316 route, and the number of the reserved time steps is equal to the length of the treatment duration.
 317 Constraints (23) and (24) connect variables θ_{djt} to ε_{ju} ; that is, therapist j 's time steps that are
 318 reserved for the treatment route are located in the time window $[\varepsilon_{ju}, \varepsilon_{ju} + d_{ju}]$.

5.3. Pricing algorithm

As described in Section 5.2, each patient is associated with a pricing subproblem SP_u , which is to find a treatment route with most negative reduced cost for this patient. To solve the pricing subproblem SP_u of each patient u , we propose a tailored pricing method, i.e. a column generator based on genetic algorithm (GA) which is known as one of metaheuristic algorithms where a population of candidate solutions to an optimization problem is evolved toward better solutions (Gendreau et al., 2002). The important components of the GA applied in this paper are described as follows.

Algorithm 3: Scheme of the GA for solving pricing subproblem SP_u

```

1 Generate an initial feasible solution;
2 Normalize the start time of treatments:  $\varepsilon_{iu}^d = \frac{\varepsilon_{iu}}{\bar{w}_i - d_{iu} - w_i}$ ;
3 Generate  $N_p - 1$  treatment routes from the initial one via mutation;
4 while Stopping criterion is not satisfied do
5     Unnormalize the start time of treatments;
6     Repair all treatment routes to satisfy constraints (6);
7     Calculate the cost for each treatment routes;
8     Select  $N_s$  best distinct treatment routes;
9     Normalize the start time of treatments;
10    Randomly pair two treatment routes;
11    for Each pair of treatment routes do
12         $d_1, d_2 \leftarrow$  The index of two treatment routes;
13         $\eta, \zeta \in [0, 1] \leftarrow$  The index of treatment and the coefficient for extrapolation method;
14         $\varepsilon_{\eta u}^{d_1} \leftarrow \varepsilon_{\eta u}^{d_1} - \zeta(\varepsilon_{\eta u}^{d_1} - \varepsilon_{\eta u}^{d_2})$ ;  $\varepsilon_{\eta u}^{d_2} \leftarrow \varepsilon_{\eta u}^{d_2} - \zeta(\varepsilon_{\eta u}^{d_2} - \varepsilon_{\eta u}^{d_1})$ ;
15        Swap the rest treatments of treatment route  $d_1$  and  $d_2$ ;
16    end
17    Calculate the total number of potential mutated treatments among all treatment routes;
18    for Each potential mutated treatment do
19         $d, i \leftarrow$  The index of treatment route and the index of treatment of selected one;
20         $\varepsilon_{iu}^d \leftarrow$  Uniformly generated from  $[0, 1]$ ;
21    end
22 end
23 Select at most  $N_c$  treatment routes with a negative reduced cost  $Z'$ ;

```

Gene coding. A solution to the pricing subproblem is coded as a single chromosome whose genes represent start times of treatments. The start time of each treatment o_{iu} is bounded in the interval $[w_i, \bar{w}_i - d_{iu}]$. For convenience, the treatments' start times are mapped into $[0, 1]$ for normalization. The normalized start time of treatment o_{iu} , denoted as ε_{iu}^n , is expressed as: $\varepsilon_{iu}^n = \frac{\varepsilon_{iu}}{\bar{w}_i - d_{iu} - w_i}$.

Initial population. In the initialization step, we generate a good initial solution from the RLMP, and then compute additional $N_p - 1$ (N_p indicating the population size) solutions via mutation operation as outlined in lines 4 to 10 of Algorithm 3.

334 *Evaluation.* The objective value, which is equal to the timespan of the corresponding treatment
 335 route, is used to measure the fitness of a chromosome. The normalized start time of treatment ε_{iu}^n
 336 is unnormalized as $\varepsilon_{iu} = \varepsilon_{iu}^n(\bar{w}_i - d_{iu} - w_i)$, and the timespan of the treatment route is calculated.
 337 When a solution (chromosome) is generated via crossover and mutation operations, the time con-
 338 sistency between patient's successive treatments (illustrated in Figure 4), i.e., constraints (6), may
 339 be violated. Therefore, a reparation step is introduced here: the overdue treatment is shifted right
 340 on time axis to obtain a feasible solution. If the solution is still not feasible after reparation, its
 341 fitness is set to a sufficiently large positive value.

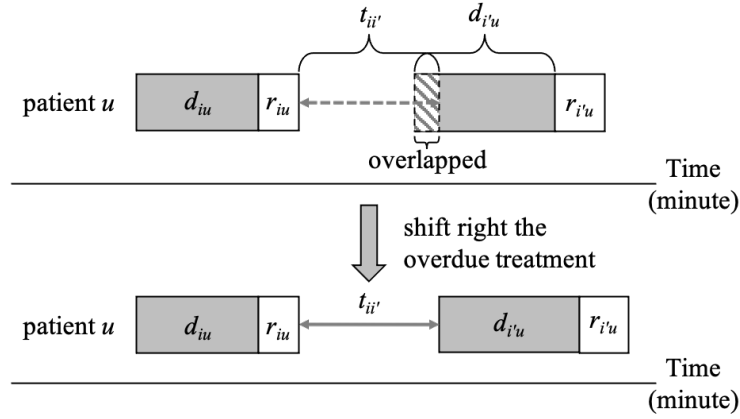


Figure 4: Reparation step

342 *Parent selection.* In each generation, a tabu-like list is considered to select distinct solutions with
 343 negative reduced cost, which is calculated by (21). Then, two solutions are randomly chosen as
 344 parent chromosomes.

345 *Crossover.* The crossover is a key enabler of continuous genetic algorithm. Haupt and Haupt
 346 (2004) proposed a crossover method combined with extrapolation to introduce new values outside
 347 the range of their parents without letting the algorithm stray too far. As illustrated in Figure 5,
 348 given two parents, randomly select one point to process extrapolation and the rest of points are
 349 then swapped between two patients as shown in Figure 6. The crossover method is summarized in
 350 lines 10–16 in Algorithm 3.

351 *Mutation.* A mutation rate of 20% is chosen for some of the variables to avoid some problems
 352 of overly fast convergence. Moreover, we randomly select the treatment and the corresponding
 353 treatment route to determine which variable is to be mutated and then a 0-1 uniformly distributed
 354 value is generated to replace the legacy one. The mutation is outlined in lines 17–21 in Algorithm
 355 3.

356 *Stopping criteria.* The GA pricing algorithm terminates when either of the following two conditions
 357 is fulfilled: (1) the incumbent reduced cost associated to the pricing problem, which is calculated by

358 objective function (21), is not improved after a fixed number I_s of iterations ; (2) overall maximum
 359 CPU time of the column generation algorithm (denoted by *Total_Time_Lim* in Algorithm 2) is
 360 reached. In our implementation, we set $I_s=3$ and the maximum CPU time at three hours.

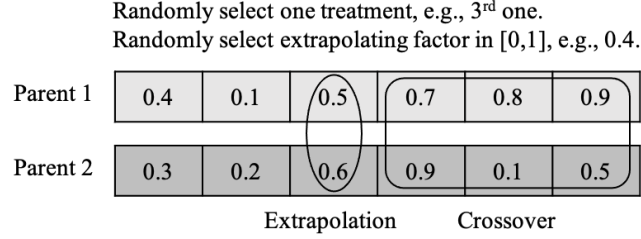


Figure 5: A pair of parents

Offspring 1 $0.5 - 0.4 * (0.5 - 0.6) = 0.54$

Offspring 2 $0.6 - 0.4 * (0.6 - 0.5) = 0.56$

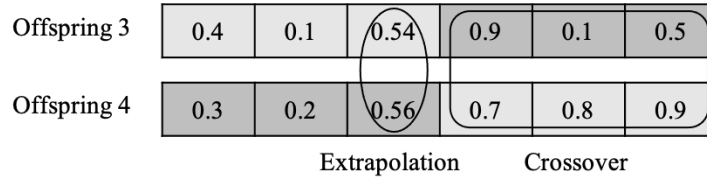


Figure 6: New offspring produced by crossover step

361 5.4. Accelerating strategies

362 Column generation is known for its poor convergence, and it may be relatively time consuming
 363 to prove optimality of a degenerate optimal solution (see, for example, Lübbecke and Desrosiers
 364 (2005)). We implement the following three accelerating strategies to speed up the column generation
 365 process.

366 5.4.1. Initialization of the RLMP

367 In the initialization step of the column generation procedure, a set of columns need to be
 368 generated for the RLMP. Even though in principle only artificial columns are sufficient, feasible
 369 ones can be used if they are available. Numerous studies have confirmed the benefits of using feasible
 370 columns to initialize the RLMP. For this reason, we use the greedy heuristic RFSS described in
 371 Section 4 to initialize the RLMP.

372 5.4.2. A Warm-start strategy for the pricing subproblem

373 As observed from algorithms 2 and 3, several subproblems are solved at each iteration of the
 374 column generation algorithm. The literature shows that an intelligent initialization or a previous
 375 solution of the RLMP can provide a *warm start* of the method (Vanderbeck, 2005; Desrosiers and

Lübbecke, 2005), and thereby to speed up the convergence of the column generation procedure. Inspired by that, a similar warm-start strategy is used to avoid the computationally costly procedure at the outset of pricing process. That is, at step 1 of Algorithm 3, we initialize the pricing subproblem based on the set of treatments routes forming the best solution associated with the last RLMP, rather than using randomly generated or artificial ones.

5.4.3. Adding multiple distinct columns

To further speed the convergence of the column generation solution process, at each iteration of the column generation procedure, we add more than one column having negative reduced cost (if any). In the computational experiments reported in the next section, we also use the general MIP solver provided by CPLEX to solve the pricing subproblems SPu. In this case, we also use the solution pool feature provided by CPLEX to examine multiple solutions (either optimal or not) to MIP models.

6. Computational results

Our numerical experiments are reported in this section. To validate the proposed methods, the compact model RPSRP defined by constraints (1)–(15) is solved by CPLEX 12.51. The results of greedy heuristic and column generation based approach are then provided. The RLMP and the final reduced MP problems involved in the column generation based approach are also solved by CPLEX. All algorithms were implemented in C++. Experiments were carried out on a server equipped with an Intel Xeon E5-2760 2.5-GHz CPU and 125 GB of RAM under CentOS Linux 7.

6.1. Test instances

The test instances are derived from daily real data of a moderate-sized rehabilitation hospital (i.e., Shanghai No. 3 Rehabilitation Hospital). They are characterized by three factors: number of therapists, number of patients, and total number of treatments, named in this order in the tables. According to the number of therapists, the instances are categorized into small-, medium- and large-scale settings. Tables 4–6 report the minimum (Min.), maximum (Max.) and average (Avg.) number of treatments per patient (Num. of Trt. per Pat.) and per therapist (per Thp.). The workload of the therapists is also summarized for each instance, and measured by the time utilization per therapist: $\sum_{u \in P_i} d_{iu} / (\bar{w}_i - w_i)$. It ranges between 35% and 75%, with an average of approximately 50% of their regular working timespan. Given the fact that the number of treatments per patient ranges from one to seven, the shortest path of each patient’s treatment route can be conveniently computed by the CPLEX solver.

Table 4: Time utilization rate of therapist in small-scale instances

Instance	Num. of Trt. per Pat.			Num. of Trt. per Thp.			Time Utilization per Thp.		
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
5-95-107	1	2	1.1	7	57	21.4	35.29%	63.79%	46.90%
10-186-242	1	3	1.3	7	77	24.2	34.45%	64.71%	43.85%
15-272-339	1	3	1.2	7	85	22.6	35.83%	73.28%	51.91%
20-327-476	1	4	1.5	7	85	23.8	35.83%	74.36%	52.84%
25-429-609	1	4	1.4	7	85	24.4	36.44%	73.73%	52.40%
30-429-702	1	4	1.6	7	85	23.4	35.29%	73.28%	46.98%
35-447-886	1	6	2.0	7	85	25.3	35.29%	73.73%	49.74%
40-471-930	1	7	2.0	7	85	23.2	33.61%	69.67%	48.54%
45-528-1045	1	6	2.0	7	85	23.2	33.33%	73.73%	49.00%
50-639-1154	1	6	1.8	7	85	23.1	34.45%	75.00%	50.97%
Avg.	1.0	4.5	1.6	7.0	81.4	23.5	34.98%	71.53%	49.31%

Table 5: Time utilization rate of therapist in medium-scale instances

Instance	Num. of Trt. per Pat.			Num. of Trt. per Thp.			Time Utilization per Thp.		
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
51-665-1158	1	6	1.7	7	85	22.7	34.75%	74.36%	47.55%
52-671-1196	1	7	1.8	7	85	23.0	34.43%	72.03%	47.62%
53-687-1241	1	7	1.8	7	85	23.4	33.88%	71.43%	47.81%
54-711-1263	1	6	1.8	7	85	23.4	34.17%	75.65%	48.58%
55-719-1295	1	6	1.8	7	77	23.5	34.15%	76.32%	47.10%
56-741-1350	1	5	1.8	7	77	24.1	34.71%	73.73%	53.17%
57-756-1396	1	6	1.8	7	85	24.5	33.88%	72.65%	50.58%
58-788-1436	1	6	1.8	7	85	24.8	34.17%	73.28%	50.51%
59-793-1508	1	5	1.9	7	85	25.6	34.75%	73.11%	50.35%
60-796-1618	1	6	2.0	7	85	27.0	34.45%	73.73%	48.27%
Avg.	1.0	6.0	1.8	7.0	83.4	24.2	34.33%	73.63%	49.15%

Table 6: Time utilization rate of therapist in large-scale instances

Instance	Num. of Trt. per Pat.			Num. of Trt. per Thp.			Time Utilization per Thp.		
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
70-824-1601	1	6	1.9	7	85	22.9	33.33%	73.73%	47.02%
80-882-1973	1	6	2.2	7	85	24.7	34.17%	72.50%	49.41%
90-916-2026	1	7	2.2	7	85	22.5	34.43%	72.50%	49.16%
100-926-2049	1	7	2.2	7	85	20.5	34.17%	73.73%	48.53%
110-994-2295	1	7	2.3	7	85	20.9	34.43%	73.11%	48.02%
120-1103-2352	1	6	2.1	7	85	19.6	33.88%	75.00%	47.83%
130-1264-2876	1	7	2.3	7	85	22.1	34.17%	73.73%	48.85%
140-1315-3115	1	7	2.4	7	85	22.2	33.33%	73.73%	49.44%
150-1471-3658	1	7	2.5	7	85	24.4	34.15%	74.36%	49.44%
160-1532-3747	1	7	2.4	7	85	23.4	34.71%	75.00%	49.94%
Avg.	1.0	6.7	2.3	7.0	85.0	22.3	34.08%	73.74%	48.77%

6.2. Computational results of solving the RPSRP model directly

Table 7 summarizes the computational results obtained by using CPLEX to solve the RPSRP model directly. A valid lower bound (LB) of the problem is proposed, by relaxing constraints (7) and (8). Detailed informations on instance size and solution quality of 30 instance are provided. The instance-size informations include the number of variables (Num. of Var.), the number of binary variables (Num. of Binary Var.), the number of constraints (Num. of Constr.), and the number of non-zeros in the constraint matrix. In the solution quality part, the best bound value (OBJ_{BB}), the LB value (OBJ_{LB}), the IP solution value (OBJ_{IP}), the IP time in seconds, the optimality gap, and the IP-LB gap are reported. We set the maximum CPU time to three hours.

As we can observe from Table 7, we can only obtain feasible solutions only for three out of 30 instances (5-95-107, 10-186-242, 15-272-339) within the three hour computation time, and only one optimal solution (instance 5-95-107). In Table 7, “-” denotes the unsolved cases. The best bound found by CPLEX is poor. Therefore, in the followin, to evaluate the performance of our proposed RFSS method and column generation based method, we use the LB values to access the quality of solutions obtained with these algorithms.

6.3. Computational results of the “route-first, schedule-second” (RFSS) algorithm

A summary of the results obtained with the proposed greedy method based on “route-first, schedule-second” (RFSS) strategy is given in Table 8. The LB values (OBJ_{LB}), the solution values of the RFSS method (OBJ_{RFSS}) and the computation time of RFSS heuristic, made up of the parts of “route” and “schedule” phases, are highlighted in Table 8. The solution quality of RFSS is identified by the IP-LB gap equal to $(OBJ_{RFSS} - OBJ_{LB})/OBJ_{RFSS}$. The numerical results clearly show that all instances can be solved efficiently (within 17.9 seconds), with an average IP-LB gap of 43.35%. More specifically, the computation time is mainly taken in “route” phase as we solve it by CPLEX directly. The obtained solution from RFSS method is not ideal in terms of IP-LB gap. However, since such a feasible solution can be rapidly provided by RFSS, it can constitute a reasonably good initial solution to warm the column generation based approach rather than using a randomly generated one.

Table 7: Numerical results for the instances solved by CPLEX directly

Instance	Model CM				LB		Best Bound		IP		Quality of IP Solution	
	Num. of Var.	Num. of Binary Var.	Num. of Constr.	Num. of Non-zeros.	OBJ _{LB}	OBJ _{BB}	OBJ _{BB}	Gap	Solution OBJ _{IP}	CPU Time (s)	Optimality Gap	IP-LB Gap
5-95-107	4713	4106	9008	25955	2835	2835	2835	0.00%	2835	248	0.00%	0.00%
10-186-242	12147	10862	23366	67968	8515	4785	4785	-77.95%	8520	10781	43.84%	0.06%
15-272-339	19394	17560	37487	109355	11085	6900	6900	-60.65%	11125	10777	37.98%	0.36%
20-327-476	27444	25034	53049	155416	16550	6985	6985	-136.94%	-	10779	-	-
25-429-609	36277	33162	70196	205709	21565	9540	9540	-126.05%	-	10780	-	-
30-429-702	33462	30068	64167	187676	26460	8157	8157	-224.38%	-	10786	-	-
35-447-886	48990	44990	94335	278286	34375	6289	6289	-446.55%	-	10787	-	-
40-471-930	48408	44204	92983	273990	35970	6435	6435	-458.97%	-	10784	-	-
45-528-1045	52746	48026	101243	298055	41220	6740	6740	-511.57%	-	10777	-	-
50-639-1154	63298	57918	121865	358792	44250	11044	11044	-300.67%	-	10774	-	-
51-665-1158	60040	54570	115507	339084	43210	9695	9695	-345.69%	-	10785	-	-
52-671-1196	65094	59492	125301	368724	46570	11490	11490	-305.31%	-	10785	-	-
53-687-1241	69197	63412	133404	392703	48150	11260	11260	-327.62%	-	10774	-	-
54-711-1263	63591	57668	122138	358573	46795	10945	10945	-327.55%	-	10786	-	-
55-719-1295	64293	58250	123270	362169	51970	13725	13725	-278.65%	-	10779	-	-
56-741-1350	61496	55222	117433	344462	49625	10565	10565	-369.71%	-	10784	-	-
57-756-1396	78611	72154	151632	446588	54170	12490	12490	-333.71%	-	10779	-	-
58-788-1436	78835	72162	151898	447078	54905	12855	12855	-327.11%	-	10777	-	-
59-793-1508	86248	79344	166385	490586	58950	11690	11690	-404.28%	-	10782	-	-
60-796-1618	88261	81018	169754	501094	66630	12490	12490	-433.47%	-	10772	-	-
70-824-1601	77094	69818	147561	434061	63400	12645	12645	-401.38%	-	10781	-	-
80-882-1973	100076	91510	191807	566621	78010	8650	8650	-801.85%	-	10779	-	-
90-916-2026	93203	84376	177860	524384	76710	8690	8690	-782.74%	-	10782	-	-
100-926-2049	92146	83220	175607	517609	80465	9845	9845	-717.32%	-	10780	-	-
110-994-2295	107498	97630	205199	605893	89745	9335	9335	-861.33%	-	10772	-	-
120-1103-2352	101950	91584	194149	570996	89725	11420	11420	-685.68%	-	10772	-	-
130-1264-2876	136435	124014	260382	769206	114490	14570	14570	-685.79%	-	10773	-	-
140-1315-3115	156309	143018	299300	885147	124665	11185	11185	-1014.57%	-	10778	-	-
150-1471-3658	195202	179814	374389	1109748	149575	13155	13155	-1037.02%	-	10763	-	-
160-1532-3747	195330	179492	374663	1109151	153005	11115	11115	-1276.56%	-	10771	-	-

Note: 'Num.' denote the total number; 'Var.' denote the variables; 'Constr.' denote the constraints; '-' denotes the unsolved cases. 'BB-LB Gap' is computed by $(\text{OBJ}_{\text{BB}} - \text{OBJ}_{\text{LB}}) / \text{OBJ}_{\text{BB}}$. 'Optimality Gap' is reported by CPLEX. 'IP-LB Gap' is computed by $(\text{OBJ}_{\text{IP}} - \text{OBJ}_{\text{LB}}) / \text{OBJ}_{\text{IP}}$.

Table 8: Numerical results for the instances solved by greedy based method (RFSS)

Instance	LB	RFSS	CPU			IP-LB Gap
	OBJ _{LB}	OBJ _{RFSS}	Route	Schedule	Total	
5-95-107	2835	5175	0.6	0.0	0.6	45.22%
10-186-242	8515	14055	1.4	0.0	1.4	39.42%
15-272-339	11085	20190	2.0	0.0	2.0	45.10%
20-327-476	16550	32405	3.5	0.0	3.5	48.93%
25-429-609	21565	39770	4.5	0.0	4.5	45.78%
30-429-702	26460	43130	5.3	0.0	5.3	38.65%
35-447-886	34375	64540	8.8	0.0	8.8	46.74%
40-471-930	35970	63840	12.4	0.0	12.4	43.66%
45-528-1045	41220	73465	9.9	0.0	9.9	43.89%
50-639-1154	44250	76190	12.1	0.0	12.1	41.92%
51-665-1158	43210	82040	10.3	0.0	10.3	47.33%
52-671-1196	46570	84210	15.4	0.0	15.4	44.70%
53-687-1241	48150	84450	16.2	0.0	16.2	42.98%
54-711-1263	46795	87870	12.4	0.0	12.5	46.75%
55-719-1295	51970	82815	14.7	0.0	14.8	37.25%
56-741-1350	49625	83045	13.7	0.0	13.7	40.24%
57-756-1396	54170	98295	13.6	0.0	13.6	44.89%
58-788-1436	54905	95375	14.9	0.1	14.9	42.43%
59-793-1508	58950	107725	15.2	0.1	15.3	45.28%
60-796-1618	66630	114755	20.3	0.1	20.4	41.94%
80-882-1973	78010	140715	24.4	0.1	24.5	44.56%
90-916-2026	76710	137760	29.4	0.1	29.5	44.32%
100-926-2049	80465	134695	21.8	0.1	21.8	40.26%
110-994-2295	89745	157395	30.6	0.1	30.7	42.98%
120-1103-2352	89725	159990	21.3	0.1	21.4	43.92%
130-1264-2876	114490	193990	39.9	0.1	40.0	40.98%
140-1315-3115	124665	226930	42.4	0.2	42.6	45.06%
150-1471-3658	149575	254710	52.9	0.2	53.1	41.28%
160-1532-3747	153005	266860	46.9	0.3	47.1	42.66%
Avg.			17.8	0.1	17.9	43.35%

6.4. Computational results of the column generation based method

We now present the results of the proposed column generation method with the GA pricing algorithm, called CG_{GA} . Problem RLMP and the final reduced MP problem are solved by CPLEX. First, we examine the GA pricing algorithm by comparing it with the direct solution of the pricing problem using CPLEX, denoted as CG_{CPLEX} . We then discuss the effects of accelerating strategies on CG_{GA} . The maximum computation time is set to three hours for the pricing subproblems and the IP master problem, respectively.

6.4.1. Evaluation of the proposed GA pricing algorithm

For both CG_{GA} and CG_{CPLEX} , we apply the accelerating strategies: initialization of the RLMP by RFSS heuristic, warm-start of the pricing subproblem, and returning multiple columns to the RLMP. In Tables 9–11, detailed results of CG_{GA} and CG_{CPLEX} are provided for different scales of instances: the number of generated columns, the number of iterations, the CPU time to solve the RLMP, the CPU time to solve the pricing subproblem, the CPU time to solve the IP master problem, the total solution time, the IP-LP gap and the IP-LB gap. It is worthy note that, for IP-LP with CG_{CPLEX} , we compute a valid lower bound since the pricing subproblem is solved to optimality (and no time limits occur); for CG_{GA} , despite the pricing subproblem is solved heuristically, the IP-LP gaps can still be used to compare the integral solutions of the MP. On the other hand, LB provides a valid lower bound. The IP-LB gaps can always be used to check if a solution is optimal or not, in both cases of CG_{CPLEX} and CG_{GA} , therefore a gap equal to zero in the tables means that the corresponding solution is solved to optimality. Furthermore, a mark ‘►’ indicates that based on the generated columns, the IP-LP gap is zero. A mark ‘*’ means that the IP-LB gap is less than 0.05%, which means that an optimal or near optimal solution of RPSRP has been found.

From Tables 9– 11, we can make the following observations:

- (1) The column generation methods with two different pricing methods can both obtain good solutions for the RPSRP, clearly outperforming the solutions of solving the RPSRP directly by CPLEX and the RFSS heuristic. We notice that for all-scale instances, the largest IP-LP gaps of CG_{GA} and CG_{CPLEX} are equal to 0.05% and 0.61%, respectively. In fact, for CG_{GA}/CG_{CPLEX} , their average IP-LP gaps are 0.00%/0.01%, 0.00%/0.05% and 0.01%/0.33% in small-, medium-, and large-scale instances. Also, the IP-LP gaps are zero for 21 and 12 out of 30 instances when we use CG_{GA} and CG_{CPLEX} .
- (2) With the increase of instance scale, the mean IP-LB gaps of CG_{GA} are equal to 0.03%, 0.10% and 0.11%, which are receptively obtained within 76.2 s, 163.8 s, 3487.7 s on average. Among all instances, the largest IP-LB gap is only 0.24%. More specifically, the IP-LB gaps are zero for six out of 30 instances. This implies that the optimal solutions of the RPSRP are found via CG_{GA} in these cases. Also, for 19 out of 30 instances, the IP-LB gaps are no greater than 0.10%. These results validate the effectiveness of the the proposed GA column generator, as well as our column generation method CG_{GA} .

(3) Concerning the comparison between the two column generation methods CG_{GA} and CG_{Cplex} , there is no significant difference in the IP-LB gaps. We find that the CG_{GA} produces about eight times more columns than CG_{Cplex} , and therefore more running time is needed to solve the RLMP, since a larger pool of columns has to be dealt with. However, the CG_{GA} decreases the time to solve the pricing subproblem and the IP solution time, which leads to significant reductions in total running time. Furthermore, CG_{Cplex} consumes considerable time to solve the IP master problem, due to the existence of a large number of similar columns in the column pool. Obviously, CG_{GA} is much more competitive than CG_{Cplex} in terms of overall computation time, which reveals that our GA column generator outperforms CPLEX as a pricing method. The superiority of GA column generator is more apparent in the large-scale settings, since the CG_{GA} provides solutions with smaller IP-LB gaps (except for instance 70-824-1601), within a much shorter time.

Table 9: Performance of the proposed column generation method CG_{GA} for small-scale instances

Instance	Solution Method	Num. of Columns	Num. of Iterations	CPU				IP-LP Gap	IP-LB Gap
				RLMP (s)	Pricing (s)	IP (s)	Total (s)		
5-95-107	CG_{GA}	2376	3	0.3	0.4	0.1	0.8	► 0.00%	* 0.00%
	CG_{Cplex}	344	7	0.3	23.1	0.0	23.4	► 0.00%	* 0.00%
10-186-242	CG_{GA}	9319	8	1.1	2.8	0.1	4.1	► 0.00%	* 0.00%
	CG_{Cplex}	1085	7	0.8	80.1	0.1	81.0	► 0.00%	* 0.00%
15-272-339	CG_{GA}	13960	3	1.5	1.1	0.6	3.2	► 0.00%	* 0.00%
	CG_{Cplex}	1627	9	1.2	151.7	0.2	153.2	► 0.00%	* 0.00%
20-327-476	CG_{GA}	19751	10	5.6	8.5	1.0	15.1	► 0.00%	* 0.00%
	CG_{Cplex}	2765	9	2.2	335.6	1.3	339.0	► 0.00%	* 0.00%
25-429-609	CG_{GA}	26502	9	9.1	9.5	2.1	20.8	► 0.00%	* 0.00%
	CG_{Cplex}	3534	9	2.9	432.6	1.2	436.7	► 0.00%	* 0.00%
30-429-702	CG_{GA}	29237	10	11.3	14.2	3.2	28.7	► 0.00%	* 0.00%
	CG_{Cplex}	3447	9	3.6	847.7	1.0	852.4	► 0.00%	* 0.00%
35-447-886	CG_{GA}	45911	10	27.2	25.4	120.4	173.0	► 0.00%	0.15%
	CG_{Cplex}	5209	13	7.0	4051.9	3011.1	7069.9	► 0.07%	0.07%
40-471-930	CG_{GA}	43828	11	41.6	25.9	16.9	84.4	► 0.00%	* 0.04%
	CG_{Cplex}	5518	15	10.4	5886.3	2148.6	8045.3	► 0.00%	* 0.01%
45-528-1045	CG_{GA}	53155	11	35.2	28.9	7.5	71.7	► 0.00%	0.11%
	CG_{Cplex}	5476	10	8.0	3106.9	5.3	3120.2	► 0.00%	* 0.00%
50-639-1154	CG_{GA}	62091	11	45.6	32.6	282.4	360.6	► 0.00%	* 0.02%
	CG_{Cplex}	6502	11	11.9	4508.8	10759.7	15280.5	0.06%	0.06%
Avg.	CG_{GA}	30613.0	8.6	17.9	14.9	43.4	76.2	0.00%	0.03%
	CG_{Cplex}	3550.7	9.9	4.8	1942.5	1592.8	3540.2	0.01%	0.01%

Table 10: Performance of the proposed column generation method CG_{GA} for medium-scale instances

Instance	Solution	Num. of Columns	Num. of Iterations	CPU				IP-LP	IP-LB
	Method			RLMP (s)	Pricing (s)	IP (s)	Total (s)	Gap	Gap
51-665-1158	CG_{GA}	55737	11	40.8	28.0	12.2	81.0	► 0.00%	0.09%
	CG_{CPLEX}	6491	9	8.3	2189.3	70.8	2268.4	► 0.00%	* 0.00%
52-671-1196	CG_{GA}	60614	13	47.1	37.7	15.6	100.4	► 0.00%	0.24%
	CG_{CPLEX}	6643	13	12.4	5467.1	633.6	6113.1	► 0.00%	* 0.00%
53-687-1241	CG_{GA}	54682	10	44.5	29.6	20.3	94.4	► 0.00%	* 0.03%
	CG_{CPLEX}	6592	11	12.3	3430.9	1867.4	5310.6	► 0.00%	* 0.00%
54-711-1263	CG_{GA}	53195	10	29.3	28.9	12.4	70.7	► 0.00%	* 0.04%
	CG_{CPLEX}	6305	7	9.0	1980.2	656.1	2645.3	► 0.04%	* 0.04%
55-719-1295	CG_{GA}	63626	12	41.9	39.6	23.0	104.6	► 0.00%	0.10%
	CG_{CPLEX}	6348	9	11.7	4047.3	2476.6	6535.5	► 0.03%	* 0.03%
56-741-1350	CG_{GA}	64605	10	42.5	35.7	214.9	293.1	► 0.00%	0.17%
	CG_{CPLEX}	7469	11	10.6	5176.8	10655.1	15842.5	0.13%	0.13%
57-756-1396	CG_{GA}	68914	11	58.7	39.7	48.1	146.6	► 0.00%	* 0.02%
	CG_{CPLEX}	7825	11	13.3	3769.3	10637.8	14420.4	0.06%	0.06%
58-788-1436	CG_{GA}	70749	11	60.5	40.9	111.4	212.8	► 0.01%	0.13%
	CG_{CPLEX}	8062	15	13.2	5753.4	59.5	5826.1	► 0.00%	* 0.00%
59-793-1508	CG_{GA}	81271	11	62.6	45.3	23.6	131.4	► 0.01%	0.06%
	CG_{CPLEX}	8227	10	14.4	4569.3	10712.6	15296.3	0.08%	0.08%
60-796-1618	CG_{GA}	83739	13	90.2	63.0	249.7	402.9	► 0.01%	0.11%
	CG_{CPLEX}	7869	11	19.6	7101.5	10702.5	17823.5	0.15%	0.15%
Avg.	CG_{GA}	65713.2	11.2	51.8	38.8	73.1	163.8	0.00%	0.10%
	CG_{CPLEX}	7183.1	10.7	12.5	4348.5	4847.2	9208.2	0.05%	0.05%

Table 11: Performance of the proposed column generation method CG_{GA} for large-scale instances

Instance	Solution	Num. of Columns	Num. of Iterations	CPU				IP-LP	IP-LB
	Method			RLMP (s)	Pricing (s)	IP (s)	Total (s)	Gap	Gap
70-824-1601	CG_{GA}	80181	12	74.1	52.5	15.6	142.2	► 0.00%	0.10%
	CG_{CPLEX}	8203	11	14.9	6199.5	10657.6	16872.1	0.09%	0.09%
80-882-1973	CG_{GA}	101393	11	127.0	68.7	486.6	682.4	► 0.01%	* 0.05%
	CG_{CPLEX}	10218	9	26.5	8267.6	10692.7	18986.9	0.29%	0.29%
90-916-2026	CG_{GA}	107938	11	149.9	68.4	1336.5	1554.8	► 0.01%	* 0.04%
	CG_{CPLEX}	10591	10	34.5	9219.1	10662.0	19915.6	0.34%	0.34%
100-926-2049	CG_{GA}	110416	15	162.8	99.3	263.5	525.5	► 0.00%	0.08%
	CG_{CPLEX}	9961	12	31.8	11198.9	10226.1	21456.8	0.18%	0.18%
110-994-2295	CG_{GA}	122552	12	221.8	94.3	1045.0	1361.1	► 0.01%	0.13%
	CG_{CPLEX}	11010	8	47.0	8360.1	10401.8	18809.0	0.46%	0.47%
120-1103-2352	CG_{GA}	131864	13	183.2	88.0	109.5	380.7	► 0.00%	0.12%
	CG_{CPLEX}	11459	7	32.7	5931.0	10678.6	16642.3	0.22%	0.23%
130-1264-2876	CG_{GA}	176013	18	341.9	185.8	1170.7	1698.4	► 0.01%	0.19%
	CG_{CPLEX}	14414	8	67.1	11341.8	10728.5	22137.4	0.61%	0.66%
140-1315-3115	CG_{GA}	180466	15	372.1	164.6	5156.8	5693.5	► 0.00%	0.12%
	CG_{CPLEX}	16422	8	68.6	10949.8	10783.8	21802.2	0.57%	0.59%
150-1471-3658	CG_{GA}	190243	12	474.1	160.6	10798.9	11433.6	0.05%	0.19%
	CG_{CPLEX}	18713	6	101.1	10950.3	10784.8	21836.2	0.34%	0.49%
160-1532-3747	CG_{GA}	188405	12	447.8	159.5	10797.7	11405.1	0.04%	0.10%
	CG_{CPLEX}	20783	7	86.9	11562.5	10778.7	22428.1	0.15%	0.18%
Avg.	CG_{GA}	138947.1	13.1	255.5	114.2	3118.1	3487.7	0.01%	0.11%
	CG_{CPLEX}	13177.4	8.6	51.1	9398.1	10639.5	20088.6	0.33%	0.35%

6.4.2. Effectiveness of accelerating strategies

In this study, two strategies are mainly used to accelerate the column generation method in process of solving the RLMP, i.e., the warm start (WS) for pricing the subproblems and adding multiple columns (AMC) at each iteration. To discuss their effects on the proposed column generation method, we test two versions of CG_{GA} in the next experiments: one without WS (denoted as “No WS”), and the other without AMC (denoted as “No AMC”). The former version means that in each column generation iteration, every pricing subproblem SP_u starts from the initial solution of the RLMP, the latter version refers that the GA column generator only returns the column with the most-negative reduced cost in each column generation iteration.

Based on the results from Table 12–Table 14, we can make the following observations:

- (1) For each scale of instances, the algorithm without WS always performs worst among three versions in terms of average IP-LB gap and overall computation time. For example, the average IP-LB gaps of the algorithm without WS of small-, medium-, and large-scale instances are 4.17%, 6.43% and 9.68%, which are much larger than those of the original CG_{GA} , i.e. 0.03%, 0.10% and 0.11%. Moreover, Tables 12–14 show that the average running times of the algorithm without WS are 3297.5 s, 8780.0 s, and 11326.3 s for different scales of instances, which are much longer than those of CG_{GA} (76.2 s, 163.8 s, 3487.7 s).
- (2) For all test instances, both the algorithm without AMC and the original CG_{GA} can obtain high-quality solutions, yet the former is worse than the latter, their largest IP-LB gaps being 0.17% and 0.24%, respectively. For small-scale instances, the IP-LB gaps of the original CG_{GA} and the algorithm without AMC are 0.03% and 0.05%; the average running times of them are 76.2 s and 37.9 s. CG_{GA} spends more time in solving the RLMP and the IP master problem. For medium- and large-scale instances, the original CG_{GA} outperforms the algorithm without AMC both in terms of solution quality and computation time. Especially, concerning the average computation time, the algorithm without AMC takes about 5.99 and 1.91 times as much time as the original one in medium- and large-scale instances.
- (3) Both strategies of WS and AMC should be adopted, and the WS has more influence than the AMC on the performance of the whole algorithm. The reason is that the WS component contributes to the diversity of columns generated by the proposed GA pricing algorithm. Although the original CG_{GA} and the algorithm without WS spend approximately the same time in the pricing process, the lack of WS results in significant increase in computation time in solving the RLMPs and the final IP master problem.

Table 12: Comparison between different pricing strategies for small-scale instances

Instance	Pricing	Num. of Columns	Num. of Iterations	CPU				IP-LP	IP-LB
	Strategy			RLMP (s)	Pricing (s)	IP (s)	Total (s)	Gap	Gap
5-95-107	CG _{GA}	2376	3	0.3	0.4	0.1	0.8	► 0.00%	* 0.00%
	No WS	2421	3	0.3	0.4	0.1	0.7	► 0.00%	* 0.00%
	No AMC	133	5	0.3	0.4	0.0	0.7	► 0.00%	* 0.00%
10-186-242	CG _{GA}	9319	8	1.1	2.8	0.1	4.1	► 0.00%	* 0.00%
	No WS	7502	4	1.2	1.5	0.2	2.9	► 0.00%	0.18%
	No AMC	452	7	0.9	2.0	0.1	2.9	► 0.00%	* 0.00%
15-272-339	CG _{GA}	13960	3	1.5	1.1	0.6	3.2	► 0.00%	* 0.00%
	No WS	14452	4	1.6	1.7	0.5	3.7	► 0.00%	0.40%
	No AMC	770	8	1.3	2.8	0.1	4.2	► 0.00%	* 0.00%
20-327-476	CG _{GA}	19751	10	5.6	8.5	1.0	15.1	► 0.00%	* 0.00%
	No WS	16172	6	3.8	5.0	1.1	9.9	► 0.00%	3.36%
	No AMC	1298	14	2.5	10.6	0.3	13.5	► 0.00%	* 0.03%
25-429-609	CG _{GA}	26502	9	9.1	9.5	2.1	20.8	► 0.00%	* 0.00%
	No WS	20920	13	8.9	12.4	5.7	26.9	► 0.15%	3.96%
	No AMC	1689	13	3.6	9.9	0.9	14.4	► 0.00%	* 0.02%
30-429-702	CG _{GA}	29237	10	11.3	14.2	3.2	28.7	► 0.00%	* 0.00%
	No WS	24224	14	11.7	19.7	21.4	52.8	► 0.30%	3.38%
	No AMC	2485	17	5.3	21.9	0.9	28.1	► 0.00%	* 0.02%
35-447-886	CG _{GA}	45911	10	27.2	25.4	120.4	173.0	► 0.00%	0.15%
	No WS	31860	21	28.0	47.2	10766.5	10841.7	0.71%	8.13%
	No AMC	3170	19	11.5	41.2	15.4	68.2	► 0.03%	0.13%
40-471-930	CG _{GA}	43828	11	41.6	25.9	16.9	84.4	► 0.00%	* 0.04%
	No WS	34675	34	37.2	80.5	10761.3	10879.0	1.26%	8.83%
	No AMC	3927	19	18.0	41.8	15.2	75.0	► 0.00%	0.15%
45-528-1045	CG _{GA}	53155	11	35.2	28.9	7.5	71.7	► 0.00%	0.11%
	No WS	34788	22	28.5	55.4	206.7	290.6	► 0.18%	7.03%
	No AMC	4178	17	12.0	40.5	22.9	75.4	► 0.00%	* 0.02%
50-639-1154	CG _{GA}	62091	11	45.6	32.6	282.4	360.6	► 0.00%	* 0.02%
	No WS	43851	25	43.3	68.1	10755.9	10867.3	0.62%	6.40%
	No AMC	4036	17	17.6	40.8	37.8	96.2	► 0.01%	0.11%
Avg.	CG _{GA}	30613.0	8.6	17.9	14.9	43.4	76.2	0.00%	0.03%
	No WS	23086.5	14.6	16.4	29.2	3251.9	3297.5	0.32%	4.17%
	No AMC	2213.8	13.6	7.3	21.2	9.4	37.9	0.00%	0.05%

Table 13: Comparison between different pricing strategies for medium-scale instances

Instance	Pricing	Num. of Columns	Num. of Iterations	CPU				IP-LP	IP-LB
	strategy			RLMP (s)	Pricing (s)	IP (s)	Total (s)	Gap	Gap
51-665-1158	CG _{GA}	55737	11	40.8	28.0	12.2	81.0	► 0.00%	0.09%
	No WS	45156	16	37.1	40.2	266.0	343.3	► 0.26%	5.11%
	No AMC	4695	18	13.5	38.8	2.8	55.1	► 0.00%	0.09%
52-671-1196	CG _{GA}	60614	13	47.1	37.7	15.6	100.4	► 0.00%	0.24%
	No WS	43222	17	36.9	48.2	10781.8	10866.9	0.39%	8.44%
	No AMC	4552	19	18.1	46.5	17.2	81.8	► 0.00%	0.19%
53-687-1241	CG _{GA}	54682	10	44.5	29.6	20.3	94.4	► 0.00%	* 0.03%
	No WS	41664	16	42.9	42.7	10757.1	10842.7	0.90%	5.95%
	No AMC	4056	18	20.3	44.3	37.8	102.3	► 0.00%	* 0.04%
54-711-1263	CG _{GA}	53195	10	29.3	28.9	12.4	70.7	► 0.00%	* 0.04%
	No WS	43901	15	27.5	41.4	445.2	514.1	► 0.21%	4.19%
	No AMC	4451	21	12.7	55.0	1.9	69.6	► 0.00%	* 0.03%
55-719-1295	CG _{GA}	63626	12	41.9	39.6	23.0	104.6	► 0.00%	0.10%
	No WS	45341	26	43.8	78.8	10754.9	10877.5	0.43%	3.53%
	No AMC	4952	19	19.2	54.4	133.2	206.8	► 0.01%	0.06%
56-741-1350	CG _{GA}	64605	10	42.5	35.7	214.9	293.1	► 0.00%	0.17%
	No WS	43952	16	36.7	54.2	10761.8	10852.7	0.83%	6.75%
	No AMC	4845	19	20.9	58.3	727.3	806.5	► 0.01%	0.11%
57-756-1396	CG _{GA}	68914	11	58.7	39.7	48.1	146.6	► 0.00%	* 0.02%
	No WS	50613	17	55.3	57.4	10754.9	10867.6	0.65%	7.19%
	No AMC	5510	18	20.8	49.6	31.7	102.1	► 0.00%	0.17%
58-788-1436	CG _{GA}	70749	11	60.5	40.9	111.4	212.8	► 0.01%	0.13%
	No WS	53214	30	51.3	102.4	10768.9	10922.6	0.75%	7.57%
	No AMC	5048	17	21.1	48.8	7506.1	7576.0	► 0.05%	0.18%
59-793-1508	CG _{GA}	81271	11	62.6	45.3	23.6	131.4	► 0.01%	0.06%
	No WS	53658	15	53.5	54.3	10693.1	10800.9	0.44%	7.62%
	No AMC	7040	23	25.4	73.1	11.3	109.7	► 0.02%	0.08%
60-796-1618	CG _{GA}	83739	13	90.2	63.0	249.7	402.9	► 0.01%	0.11%
	No WS	65406	31	78.8	147.5	10685.7	10912.0	1.10%	7.97%
	No AMC	6751	21	33.8	85.2	586.6	705.6	► 0.01%	0.12%
Avg.	CG _{GA}	65713.2	11.2	51.8	38.8	73.1	163.8	0.00%	0.10%
	No WS	48612.7	19.9	46.4	66.7	8666.9	8780.0	0.60%	6.43%
	No AMC	5190.0	19.3	20.6	55.4	905.6	981.6	0.01%	0.11%

Table 14: Comparison between different pricing strategies for large-scale instances

Instance	Pricing strategy	Num. of Columns	Num. of Iterations	CPU				IP-LP	IP-LB
				RLMP (s)	Pricing (s)	IP (s)	Total (s)	Gap	Gap
70-824-1601	CG _{GA}	80181	12	74.1	52.5	15.6	142.2	► 0.00%	0.10%
	No WS	60996	35	65.2	150.5	10731.4	10947.0	0.51%	8.18%
	No AMC	6023	19	26.5	67.2	20.9	114.6	► 0.01%	0.11%
80-882-1973	CG _{GA}	101393	11	127.0	68.7	486.6	682.4	► 0.01%	* 0.05%
	No WS	73723	38	98.4	239.3	10522.6	10860.4	1.09%	8.21%
	No AMC	9294	20	47.0	98.2	3930.0	4075.2	► 0.01%	0.17%
90-916-2026	CG _{GA}	107938	11	149.9	68.4	1336.5	1554.8	► 0.01%	* 0.04%
	No WS	79404	50	145.7	325.6	10268.6	10739.9	1.37%	9.59%
	No AMC	9276	22	63.7	124.0	10298.8	10486.5	0.04%	0.10%
100-926-2049	CG _{GA}	110416	15	162.8	99.3	263.5	525.5	► 0.00%	0.08%
	No WS	74654	30	139.6	194.7	10261.6	10595.9	2.00%	9.75%
	No AMC	8818	22	56.0	116.0	10233.1	10405.2	0.08%	0.27%
110-994-2295	CG _{GA}	122552	12	221.8	94.3	1045.0	1361.1	► 0.01%	0.13%
	No WS	98737	68	195.5	521.5	10666.2	11383.2	1.98%	10.03%
	No AMC	11519	23	77.5	142.5	273.3	493.3	► 0.01%	0.12%
120-1103-2352	CG _{GA}	131864	13	183.2	88.0	109.5	380.7	► 0.00%	0.12%
	No WS	85478	32	145.8	210.8	10649.4	11006.0	1.06%	9.05%
	No AMC	11780	27	59.2	154.8	25.2	239.2	► 0.00%	0.07%
130-1264-2876	CG _{GA}	176013	18	341.9	185.8	1170.7	1698.4	► 0.01%	0.19%
	No WS	119494	47	229.3	434.8	10785.5	11449.6	2.14%	10.62%
	No AMC	13621	23	133.4	207.6	9205.8	9546.8	► 0.01%	0.17%
140-1315-3115	CG _{GA}	180466	15	372.1	164.6	5156.8	5693.5	► 0.00%	0.12%
	No WS	134874	58	316.3	630.7	10796.6	11743.6	1.41%	10.52%
	No AMC	16092	24	134.5	221.7	10797.0	11153.2	0.07%	0.24%
150-1471-3658	CG _{GA}	190243	12	474.1	160.6	10798.9	11433.6	0.05%	0.19%
	No WS	177626	93	419.2	1278.6	10797.1	12494.9	2.30%	11.17%
	No AMC	17965	24	168.0	264.9	10799.0	11231.9	0.11%	0.22%
160-1532-3747	CG _{GA}	188405	12	447.8	159.5	10797.7	11405.1	0.04%	0.10%
	No WS	174100	68	359.2	889.0	10793.8	12042.0	1.76%	9.65%
	No AMC	19409	24	175.0	275.4	8340.8	8791.2	► 0.01%	0.18%
Avg.	CG _{GA}	138947.1	13.1	255.5	114.2	3118.1	3487.7	0.01%	0.11%
	No WS	107908.6	51.9	211.4	487.6	10627.3	11326.3	1.56%	9.68%
	No AMC	12379.7	22.8	94.1	167.2	6392.4	6653.7	0.03%	0.16%

7. Conclusions

In this paper, we investigated a rehabilitation patient scheduling and routing problem. For the problem, we described compact and set partitioning-based mathematical formulations.

To solve the problem, we designed a greedy heuristic and a column generation-based heuristic. The column generation-based heuristic relies on an effective genetic algorithm used to solve the pricing problem associated with the set partitioning-based formulation. In addition, accelerating strategies are used to speed up the column generation procedure.

The proposed solution algorithms were extensively tested on a set of benchmark instances based on real data obtained from a rehabilitation hospital located in Shanghai. The results obtained show the effectiveness of the proposed solution algorithms, in particular of the column generation-based heuristic, in producing high quality solutions for real-size instances in a limited amount of computing time. An experimental analysis is also reported to attest the effectiveness of the different algorithms' components.

We see potential for further strengthening of our algorithms in the development of effective (exact) pricing algorithms for the set partitioning-based mathematical formulation based on dynamic programming techniques. In addition, we aim to investigate ways to strengthen the lower bounds derived from the formulation by means of additional valid inequalities, and to embed them into a branch-price-and-cut solution framework.

References

- Ahmadi-Javid, A., Jalali, Z., Klassen, K.J., 2017. Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research* 258, 3–34.
- Ahmadian, M.M., Khatami, M., Salehipour, A., Cheng, T., 2021. Four decades of research on the open-shop scheduling problem to minimize the makespan. *European Journal of Operational Research* .
- Balas, E., Simonetti, N., Vazacopoulos, A., 2008. Job shop scheduling with setup times, deadlines and precedence constraints. *Journal of Scheduling* 11, 253–262.
- Cayirli, T., Veral, E., 2003. Outpatient scheduling in health care: a review of literature. *Production and Operations Management* 12, 519–549.
- Chien, C.F., Huang, Y.C., Hu, C.H., 2009. A hybrid approach of data mining and genetic algorithms for rehabilitation scheduling. *International Journal of Manufacturing Technology and Management* 16, 76–100.
- Chien, C.F., Tseng, F.P., Chen, C.H., 2008. An evolutionary approach to rehabilitation patient scheduling: A case study. *European Journal of Operational Research* 189, 1234–1253.
- Desrosiers, J., Lübbecke, M.E., 2005. A primer in column generation, in: *Column generation*. Springer, pp. 1–32.

- 550 Fikar, C., Hirsch, P., 2017. Home health care routing and scheduling: A review. *Computers &*
551 *Operations Research* 77, 86–95.
- 552 Focacci, F., Lodi, A., Milano, M., 2002. A hybrid exact algorithm for the TSPTW. *INFORMS*
553 *Journal on Computing* 14, 403–417.
- 554 Gendreau, M., Laporte, G., Potvin, J.Y., 2002. Metaheuristics for the capacitated vrp, in: *The*
555 *vehicle routing problem*. SIAM, pp. 129–154.
- 556 Glover, F., Gutin, G., Yeo, A., Zverovich, A., 2001. Construction heuristics for the asymmetric
557 TSP. *European Journal of Operational Research* 129, 555–568.
- 558 Grenouilleau, F., Legrain, A., Lahrichi, N., Rousseau, L.M., 2019. A set partitioning heuristic for
559 the home health care routing and scheduling problem. *European Journal of Operational Research*
560 275, 295–303.
- 561 Guéret, C., Prins, C., 1999. A new lower bound for the open-shop problem. *Annals of Operations*
562 *Research* 92, 165–183.
- 563 Gupta, D., Denton, B., 2008. Appointment scheduling in health care: Challenges and opportunities.
564 *IIE transactions* 40, 800–819.
- 565 Hamid, M., Nasiri, M.M., Werner, F., Sheikahmadi, F., Zhalechian, M., 2019. Operating room
566 scheduling by considering the decision-making styles of surgical team members: a comprehensive
567 approach. *Computers & Operations Research* 108, 166–181.
- 568 Haupt, R.L., Haupt, S.E., 2004. *Practical Genetic Algorithms*. John Wiley & Sons.
- 569 Huynh, N.T., Huang, Y.C., Chien, C.F., 2018. A hybrid genetic algorithm with 2d encoding for
570 the scheduling of rehabilitation patients. *Computers & Industrial Engineering* 125, 221–231.
- 571 Jørgensen, H.S., Nakayama, H., Raaschou, H.O., Olsen, T.S., 1995. Recovery of walking function in
572 stroke patients: the copenhagen stroke study. *Archives of Physical Medicine and Rehabilitation*
573 76, 27–32.
- 574 Lübbecke, M.E., Desrosiers, J., 2005. Selected topics in column generation. *Operations Research*
575 53, 1007–1023.
- 576 Marynissen, J., Demeulemeester, E., 2019. Literature review on multi-appointment scheduling
577 problems in hospitals. *European Journal of Operational Research* 272, 407–419.
- 578 Samudra, M., Van Riet, C., Demeulemeester, E., Cardoen, B., Vansteenkiste, N., Rademakers, F.E.,
579 2016. Scheduling operating rooms: achievements, challenges and pitfalls. *Journal of Scheduling*
580 19, 493–525.
- 581 Savelsbergh, M.W.P., 1985. Local search in routing problems with time windows. *Annals of*
582 *Operations research* 4, 285–305.

- 583 Schimmelpfeng, K., Helber, S., Kasper, S., 2012. Decision support for rehabilitation hospital
584 scheduling. *OR Spectrum* 34, 461–489.
- 585 Vanderbeck, F., 2005. Implementing mixed integer column generation, in: *Column generation*.
586 Springer, pp. 331–358.
- 587 Xiao, L., Dridi, M., Hajjam El Hassani, A., Lin, W., Fei, H., 2018. A solution method for treatment
588 scheduling in rehabilitation hospitals with real-life requirements. *IMA Journal of Management*
589 *Mathematics* 30, 367–386.
- 590 Zhao, L., Chien, C.F., Gen, M., 2015. A bi-objective genetic algorithm for intelligent rehabilitation
591 scheduling considering therapy precedence constraints. *Journal of Intelligent Manufacturing* ,
592 1–16.

Dear Editors:

We would like to submit the enclosed manuscript entitled “A column generation-based heuristic for a rehabilitation patient scheduling and routing problem”, which we wish to be considered for publication in “**Computers and Operations Research**”. No conflict of interest exists in the submission of this manuscript, and manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

In this work, we investigated a rehabilitation patient scheduling and routing problem. For the problem, we described compact and set partitioning-based mathematical formulations. To solve the problem, we designed a greedy heuristic and a column generation-based heuristic. The column generation-based heuristic relies on an effective genetic algorithm used to solve the pricing problem associated with the set partitioning-based formulation. In addition, accelerating strategies are used to speed up the column generation procedure. The proposed solution algorithms were extensively tested on a set of benchmark instances based on real data obtained from a rehabilitation hospital located in Shanghai. The results obtained show the effectiveness of the proposed solution algorithms, in particular of the column generation-based heuristic, in producing high quality solutions for real-size instances in a limited amount of computing time. An experimental analysis is also reported to attest the effectiveness of the different algorithms' components. We hope this paper is suitable for “**Computers and Operations Research**”.

We deeply appreciate your consideration of our manuscript, and we look forward to receiving comments from the reviewers. If you have any queries, please don't hesitate to contact us.

Thank you and best regards,
Chenghao Wang