Solving the patient admission scheduling problem using constraint aggregation

Haichao Liu [a,b], Yang Wang [a], Jin-Kao Hao [b,*]

[a] *School of Management, Northwestern Polytechnical University, 127 Youyi West Road, 710072 Xi'an, China*

[b] *LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers, France*

European Journal of Operational Research, February 2024

Abstract

Patient admission scheduling (PAS) consists of assigning patients to beds over a planning horizon to maximize treatment efficiency, patient satisfaction, and hospital utilization while meeting all necessary medical constraints and considering patient preferences as much as possible. There are several different variants of the PAS problem in the literature, which differ mainly in the constraints that must be satisfied (hard) or can be violated (soft). Due to the intrinsic difficulty of the PAS problem, solving large integer programming (IP) models to optimality is challenging. In this paper, we consider the widely studied variant of the PAS problem that has the maximum number of soft constraints, and focus on how to reduce the size of IP formulations of the PAS problem to improve the solving efficiency. We employ a two-stage optimization method where the first stage builds reduced models by constraint aggregation to improve the typical formulation of the PAS problem. Experimental results on the 13 benchmark instances in the literature indicate that our method can obtain new improved solutions (new upper bounds) for 6 instances, including one proven optimal solution. For the 5 other instances whose optimal solutions are known, our approach can reach these known optimal solutions in a shorter computation time compared to the existing methods. In addition, we apply our method to the original PAS problem, which has the maximum number of hard constraints, and perform computational experiments on the same 13 benchmark instances. Our method yields 5 new best solutions and proves optimality for 6 instances.

Keywords: OR in healthcare; Patient admission scheduling; Constraint aggregation; Integer programming; Healthcare management

* Corresponding author.
Email address: `jin-kao.hao@univ-angers.fr` (Jin-Kao Hao).

1 Introduction

The demand for high-quality health services continues to increase year after year, while hospitals encounter more and more difficulties in terms of limited medical resources (Zhang et al. (2021)). Among the numerous difficulties faced, hospital admission management is a fundamental challenge of many hospital departments. One key issue is to optimize the use of bed resources as much as possible, given that beds are a critical and limited resource in a hospital (Litvak & Bisognano (2011)). An effective bed management is also essential to promote the successful flow of patients through a health service.

The Patient Admission Scheduling (PAS) problem, first introduced by Demeester et al. (2008), consists of assigning patients (whose admission dates and length of stay are known in advance) to beds in specific departments on each day of their hospitalization while satisfying a number of hard constraints and as many soft constraints as possible. Over the past few years, this problem has received increasing attention in the literature. As a result, several different variants of the PAS problem have been proposed, differing mainly in the treatment of some constraints as hard or soft. These constraints include gender policy, age policy, mandatory equipment, single room requirement, and patient transfer. In the literature, a widely studied variant of the PAS problem, which we refer to as the standard PAS problem, considers these constraints as soft constraints (Ceschia & Schaerf, 2011; Bastos et al., 2019). Moreover, various optimization approaches, both heuristic and exact methods, have been proposed to address this problem. The most successful heuristic methods include simulated annealing-based metaheuristics (Ceschia & Schaerf, 2011, 2012, 2016; Lusby et al., 2016) and mixed integer programming (MIP)-based matheuristics (Range et al., 2014; Turhan & Bilgen, 2017; Guido et al., 2018). These heuristics can provide high-quality solutions in a short time. However, the best results on most benchmark instances have been achieved using exact mathematical programming techniques, as reported in Bastos et al. (2019). Nevertheless, the MIP model proposed in Bastos et al. (2019) becomes too large to be solved to optimality in a reasonable time for large instances.

The integer programming (IP) model of Ceschia & Schaerf (2011) is more compact, but it is still too large to be solved to optimality on most large benchmark instances. A promising way to improve the efficiency of the solution process is to reduce the size of this model. Constraint aggregation (CA) can help reduce the number of constraints of the optimization model, which is a widely used technique in mathematical programming (Trapp & Prokopyev, 2015; Benchimol et al., 2012). Based on this idea, we focus on how better formulations – in this case, through reducing the IP model of Ceschia & Schaerf (2011) by aggregating constraints – can help further improve the efficiency in solving the IP model of the PAS problem.

The contribution of this paper is twofold. First, we propose two aggregated gender policy constraints and one aggregated patient transfer constraint to reduce the size of the IP model of Ceschia & Schaerf (2011), and evaluate the effectiveness of the proposed aggregated constraints through computational experiments. Second, we apply a two-stage optimization approach using the reduced IP models to obtain optimal solutions. Specifically, for the standard PAS problem, we generate new best solutions for 6 out of the 13 benchmark instances commonly used in the literature, including one with proven optimality. Moreover, we prove the optimality of the solutions for the remaining 5 instances in a short time. Additionally, for the original PAS problem, using the same 13 benchmark instances, we obtain 5 new best solutions, 6 new best lower bounds, and proven optimality of solutions for 6 instances.

The rest of the paper is organized as follows: Section 2 provides a review of the relevant literature on the PAS problem and the CA technique. Section 3 presents the definition of the standard PAS problem and the mathematical model. Section 4 describes our solution method. Section 5 presents computational results of our IP formulations and comparisons with state-of-the-art results. Section 5 also describes our solution method for the original PAS problem and reports the computational results. Finally, conclusions and future research opportunities are addressed in Section 6.

2 Literature Review

This section presents an overview of existing works in the literature related to the PAS problem and constraint aggregations techniques.

2.1 Patient Admission scheduling problem

The PAS problem has undergone multiple extensions over the years and can be classified into static and dynamic variants. In the static variants, only elective patients are considered and all patient admission and discharge requirements are assumed deterministic. Additionally, patient admissions in these static variants are scheduled in advance. The primary difference in studies that focus on static variants lies in their treatment of soft constraints that can be violated at the cost of incurring a penalty. These constraints include gender policy, age policy, mandatory equipment, single-room requirements, and patient transfer. In the original PAS problem only the patient transfer constraint was described as a soft constraint, while the first 4 constraints were described as hard constraints, which are not allowed to be violated. However, Demeester et al. (2010) also considered these 4 constraints as soft constraints. As a result, only Range et al. (2014); Hammouri & Alweshah (2017); Guido et al. (2018) treated the first 4 constraints or part of them as hard constraints, while others considered them as soft constraints or ignored some of them. Moreover, except for Ceschia & Schaerf (2011); Range et al. (2014); Turhan & Bilgen (2017); Bastos et al. (2019), most studies treated the patient transfer constraint as a hard constraint. Considering patient transfer as a hard constraint can simplify the problem by narrowing the search space, but it may also result in only finding sub-optimal solutions for those variants that consider patient transfer as a soft constraint.

The dynamic variants (DPAS), also known as the PASU (U for uncertainty), extend the static PAS problem by considering several real-world features, such as the presence of urgent and emergency patients whose arrival dates are uncertain, uncertainty in the length of stay, and the possibility of delayed admissions (Ceschia & Schaerf, 2016; Zhu et al., 2019; Guido, 2023). The above uncertainty information is gradually revealed on a day-to-day basis, so that the DPAS problem is solved through the use of daily rescheduling. Similar to the static variants, the main difference among the dynamic variants is that some soft constraints (e.g. age policy, mandatory equipment, department specialism) are considered as hard constraints. Furthermore, in order to make the problem suitable for practical applications, some studies considered more realistic constraints such as constraints related to operating room scheduling (Ceschia & Schaerf, 2016; Zhu et al., 2019).

The PAS problem is known to be NP-hard (Vancroonenburg et al. (2011)). As a result, solving the problem is computationally challenging. In the following sections, we provide a comprehensive review of the solution approaches proposed in the literature for both the static and dynamic PAS problems, including heuristic and exact methods. Table 1 provides a summary of the existing research on the PAS problem along with the problem type, problem constraints and solution approach. We indicate both the type of PAS problem-static or dynamic-as well as specify which constraints are considered as hard or soft in each study. The symbols "✓" and "-" are used to respectively indicate the problem type and absence of the optimization model. Moreover, Section 3.3 provides detailed differences of IP/MIP models among those proposed in the literature for the static PAS problem.

2.1.1 Heuristic methods

Heuristic algorithms for solving the PAS problem aim to find good-enough solutions in a reasonable time. Existing heuristic algorithms are based either on single-trajectory search or population-based search. Among the single-trajectory search, Demeester et al. (2010) proposed an IP model to assign patients to rooms while allowing violations of some soft constraints. They ap-

Table 1
Summary of the PAS research

Reviewed Literature	Problem type		Problem constraints		Solution approach	
	Static	Dynamic	Hard	Soft	Optimization model	Algorithm
Heuristic methods						
Demeester et al. (2010)	✓		1-8	(5-8)*, 9-13	IP	H-TS
Ceschia & Schaerf (2011)	✓	✓	1-4, 14	5-13	IP	SA
Ceschia & Schaerf (2012)		✓	1, 2-3, 6-7, 10, 14	5, 9, 12-14	IP	SA
Bilgin et al. (2012)	✓		1-4, 13**	5, 7-9, 11-12	MINLP	H-H
Hammouri & Alrifal (2014)	✓		1-4, 13**	5-12	-	BBO
Range et al. (2014)	✓		1-7	9-13	IP	CG
Kifah & Abdullah (2015)	✓		1-4, 13**	5-12	MINLP	ANLGD
Ceschia & Schaerf (2016)		✓	1, 3-4, 14	5-7, 9-14	-	SA
Lusby et al. (2016)		✓	1, 3-4, 6-7, 10, 14	5, 9, 12-14	MIP	ALNS
Hammouri & Alweshah (2017)	✓		1-5, 13**	7-12	-	BBO-GBS
Turhan & Bilgen (2017)	✓		1-4	5-13	IP	F&R, F&O
Abu Doush et al. (2018)	✓		1-4, 13	5-7, 9-12	MINLP	HS
Guido et al. (2018)	✓		1-7, 13	5(partly soft), 6-7, 9-12	MIP	FiNeMath
Bolaji et al. (2018)	✓		1-4, 13**	5, 7-12	MINLP	LAHC
Bolaji et al. (2022)	✓		1-4, 13**	5, 7-12	MINLP	ABC
Hammouri (2022)	✓		1-4, 13	5-7, 9-10, 12	MINLP	MBBO-GBS
Abdalkareem et al. (2022)	✓		1-4, 13**	5-7, 9-10, 12	MINLP	DFP
Guido (2023)		✓	1, 3-4, 6-7, 14	5, 9-10, 12-14	MIP	FiNeMath-PASU
Exact methods						
Vancroonenburg et al. (2016)		✓	1, 2-3, 6-7, 10, 14	5, 9, 12-14	IP	MIP solver
Bastos et al. (2019)	✓		1-4	5-13	MIP	WS
Zhu et al. (2019)		✓	1, 3-4, 14	5-7, 9-14	MIP	MIP solver
This paper	✓		1-4	5-13	IP	WS, CA

Problem constraints: 1 - complete assignment; 2 - unchangeable date; 3 - continuous schedule; 4 - non-overlapping allocation; 5 - gender policy; 6 - age policy; 7 - mandatory equipment; 8 - single room requirement; 9 - room type preference; 10 - departmental specialism; 11 - room specialism priority; 12 - perferred room properties; 13 - patient transfer; 14 - others
* The constraints 5-8 are also incorporated into the objective function as penalties in their H-TS algorithm.
** Although the author describe the patient transfer constraint as 'soft', however, they do not provide mechanisms for transferring patients in their method. Thus, the patient transfer constraint would never be violated in their method.

plied Hybrid Tabu Search (H-TS) algorithm blended with a token-ring and a variable neighborhood descent procedure. They generated and made publicly available a set of 13 realistic benchmark instances for the PAS problem, which were largely adopted in the literature. Ceschia & Schaerf (2011) proposed an IP model, considering all soft constraints, to compute various lower bounds. Moreover, a simulated annealing (SA) algorithm is developed, which significantly improved the previous upper bounds.

Bilgin et al. (2012) proposed a hyper-heuristic (H-H) involving multiple heuristic selection and move acceptance criteria. Range et al. (2014) proposed a column generation-based (CG) heuristic, which decomposes the PAS problem into a set-partitioning problem as the master problem, and a set of room scheduling problems as the pricing problem. Kifah & Abdullah (2015) proposed an adaptive non-linear great deluge (ANLGD) algorithm, which accepts worse solutions of satisfying a given threshold. Turhan & Bilgen (2017) utilized the IP model developed by Ceschia & Schaerf (2011) and proposed two mixed integer programming-based heuristics, namely Fix-and-Relax (F&R) and Fix-and-Optimize (F&O), to obtain solutions with optimality gaps of 5-15% in less than three minutes. Bolaji et al. (2018) introduced a late acceptance hill climbing (LAHC) algorithm, which first generates an initial feasible solution and then iteratively improves the solution by applying a local search procedure. Guido et al. (2018) developed three IP models and proposed a matheuristic FiNeMath, combining the F&O heuristic, neighborhood search, and IP solvers. Their method was able to produce good results for all benchmark instances presented in Demeester et al. (2010).

For population-based methods, Hammouri & Alrifal (2014) first reported the biogeography based optimization (BBO) algorithm for the static PAS problem, which failed to improve the state-of-the-art. Later on, to improve the performance of the algorithm, the authors proposed a BBO algorithm with guided bed selection mechanism (BBO-GBS) (Hammouri & Alweshah, 2017), and a modified BBO algorithm with guided bed selection mechanism (MBBO-GBS) (Hammouri, 2022). Moreover, several researchers have attempted to improve the performance of population-based algorithms for tackling the static PAS problem, including Harmony search (HS) algorithm (Abu Doush et al., 2018), artificial bee colony (ABC) algorithm (Bolaji et al., 2022), discrete flower pollination (DFP) algorithm (Abdalkareem et al., 2022). However, these algorithms were unable to produce competitive results on the benchmark instances.

Apart from the research mentioned above, only a few studies addressed the dynamic PAS problem. Ceschia & Schaerf (2011) first introduced a dynamic case of the PAS problem in which admission and discharge dates are uncertain. They adapted their SA algorithm to solve this problem. Later, Ceschia & Schaerf (2012) formally introduced the dynamic PAS problem to account for uncertain length of stay, admission delays, and non-elective patients. This variant was solved using the SA algorithm and subsequently extended to incorporate operating room resources (Ceschia & Schaerf, 2016). Lusby et al. (2016) developed an adaptive large neighborhood search (ALNS) procedure combined with the SA framework. Their method showed superior results compared to the method suggested by Ceschia & Schaerf (2012) in most cases. Recently, Guido (2023) proposed an optimization model that plans patient admissions and patient stays considering fluctuations, and does not allow overcrowded rooms, as typically required in real-world cases. They proposed a matheurisitic FiNeMath-PASU, which is based on the FiNeMath (Guido et al., 2018).

2.1.2 Exact methods

In addition to the heuristic algorithms reviewed previously, Bastos et al. (2019) studied an exact method to solve the static PAS problem exactly. To the best of our knowledge, this is the only existing exact algorithm for the static PAS problem. The method was based on a new mathematical model, which incorporated all restrictions from the original model of Demeester et al. (2010), and applied a warm start (WS) approach to solve it with the maximum running time set to 24 hours. They reported new best upper bounds for 9 out of the 13 benchmark instances introduced in Demeester et al. (2010). Note that while Range et al. (2014), Turhan & Bilgen (2017), and Guido et al. (2018) incorporated MIP formulations into heuristic methods, these methods only improved the bounds of the optimal solution and failed to find optimal solutions.

For the dynamic PAS problem, Vancroonenburg et al. (2016) developed two IP models and considered the impact of emergency patients and patient length of stay estimates. Zhu et al. (2019) studied the compatibility of short term and long term objectives in the dynamic PAS problem, and developed multiple MIP formulations, which were solved by a MIP solver. Their approach was shown to be significantly better than the available results for 26 out of 30 benchmark instances introduced in Ceschia & Schaerf (2016).

2.2 Constraint Aggregation technique

Using CA can reduce the number of constraints of the optimization model, thereby simplifying its formulation and reducing its computational complexity. Specifically, CA involves replacing original constraints with a set of aggregated constraints, which are linear combinations of the original constraints by multipliers (Ram et al., 1988). Note that the aggregated constraints are a relaxation of the original constraints, which means that the solution space of the original constraints is a subset of the solution space of the aggregated constraints. Choices of the multipliers directly affect the strength of the aggregated constraints. In addition, the CA can suffer from poor performance when the aggregated constraints have very large coefficients, either in scale (Poirion, 2019) or in numerical values (Khurana & Murty, 2012). Thus, a crucial question of CA is how to determine the multipliers of the aggregated constraints to produce the strongest possible constraints. In this regard, researchers have proposed multiple methods such as aggregation of diophantine equations, irrational multipliers method, maximum entropy method, P-norm method, etc (Alidaee, 2014).

It is worth noting that the above CA technique, also known as static constraint aggregation (SCA) (Trapp & Prokopyev, 2015; Ermoliev et al., 1997; Rogers et al., 1991; Ram et al., 1988), aggregates constraints before the optimization process. In contrast to SCA, dynamic constraint aggregation (DCA) (Porumbel & Clautiaux, 2017; Range et al., 2014; Benchimol et al., 2012; Elhallaoui et al., 2005), in which aggregated constraints contain a subset of solutions to the original constraints, dynamically aggregates constraints during the solution process to obtain the optimal solutions. Moreover, DCA is always implemented within a column generation algorithm to solve a large set partitioning problem. The CA has been applied with success to a variety of optimization problems including multicommodity transportation (Evans, 1983), wing aero-structural optimization (Zhang et al., 2019), integrated airline crew scheduling (Saddoune et al., 2011), and set partitioning Elhallaoui et al. (2008, 2005). For a comprehensive survey of CA in optimization, see Alidaee (2014); Glover (2003); Rogers et al. (1991). In this paper, we concentrate on SCA to obtain reduced IP models for the PAS problem.

3 Problem description and mathematical model

In this section, we provide a detailed description of the standard static PAS problem and its mathematical model.

3.1 Problem constraints

The PAS problem (Demeester et al. (2010)) aims to assign patients to a set of beds during patients' hospitalizations within a given planning horizon, where the preference and the requirement of each patient are assumed to be known in advance. Specifically, each patient p has an admission date AD_p when a room is assigned to this patient and a discharge date DD_p when this patient is released from the medical treatment. The *Length of Stay* (LOS) of each patient is the duration between the admission and the discharge dates, which is expressed in nights. Patients who stay at least one night, termed *elective patient*, are eligible to be scheduled. Patients pursue medical treatments during their hospitalization, termed *specialisms*. Most of the patients need one single specialism for their entire treatment. Only a small number of patients need more than one specialism, termed *multi-spec*. Each patient is assigned to a *bed* and each *bed* belongs to a *room*. One of the most important features of the room is the *gender policy*. Rooms that require patients to be of the same gender enforce policy M (only Male) or policy F (only Female). In contrast, rooms where both genders are allowed enforce policy N (mixed gender) or policy D (on any given night, only patients of the same gender are allowed, and the gender is defined by the first patient to be scheduled in that room). There are three types of room capacity: single (one bed), twin (two beds) and ward (four beds), and each patient has a preference for a certain type of room, termed *room preference*. Each room has a different available equipment, such as oxygen and telemetry, termed *room properties*. Patients may require or prefer to be allocated to a room with the specific equipment depending on their treatment. Each room belongs to a *department* and each department is correlated with the specialisms they offer. Moreover, each department and room has its own priority degree for those specialisms. Patients must be treated at the departments where the specialism they need is offered. Each department has an *age policy* which imposes a maximum or minimum age limit for acceptance. Patients can change rooms during their hospitalization, termed $transfers$.

Based on the above problem definition, a solution is feasible if all patients are assigned to a bed such that no hard constraint of types HC1 - HC4, given below, is violated.

HC1: During the planning period, each patient must be assigned to a room.
HC2: Admission and discharge days for each patient can not be adjusted.
HC3: Patient LOS is continuous, and a patient is scheduled until his/her discharge date.
HC4: Beds allocated to patients should not overlap on any given night.

The quality of a feasible solution depends on the satisfaction of 9 types of soft constraints. If a soft constraint is violated by a solution, a penalty (a positive integer) is induced. These soft constraints SC1 - SC9 are defined as follows.

SC1: Gender policy is satisfied for each room.
SC2: Age policy is satified for each room.
SC3: A patient is assigned to a room with the required room properties for his/her treatment.
SC4: Some patients is allocated to a single room due to clinical reasons.
SC5: The room type preference for each patient is met.
SC6: A patient is allocated to a department that attends to his/her specialism.
SC7: A patient is allocated to a room that attends to his/her specialism in the first degree of priority.
SC8: A patient is assigned to a room with his/her preferred room properties.
SC9: Transfers should not be allowed.

3.2 Objective

The optimization objective of the PAS problem is to find a feasible assignment satisfying constraints HC1 - HC4 while minimizing a weighted sum of all the penalties of the unsatisfied soft constraints SC1 - SC9 (see Table 2). Formally, let Ω be the set of all feasible solutions (patient-to-bed assignments). For each $\boldsymbol{x} \in \Omega$, its cost is defined by:

$$Z(\boldsymbol{x}) = \sum_{i=1}^{9} W_i \cdot V_i(\boldsymbol{x}) \tag{1}$$

where $V_i(\boldsymbol{x})$ represents the number of times the i-th soft constraint is violated in solution $\boldsymbol{x}$, and W_i is the penalty weight corresponding to that soft constraint. The value of W_i are given in Table 2. Thus, the goal of the PAS problem is to find a feasible solution $\boldsymbol{x}^*$ such that for all $\boldsymbol{x} \in \Omega$, $Z(\boldsymbol{x}^*) \leq Z(\boldsymbol{x})$.

Table 2
Weights of the constraints.

Constraint	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9
Weight	5	10	5	10	0.8	1	1	2	11

3.3 Mathematical model

In this section, we first compare the differences of optimization models among those proposed in the literature for the static PAS problem. Taking into account the fact that beds in the same room have indistinguishable features and constraints, researchers generally formulate the PAS problem as a patient-room assignment (PRA) problem, which involves assigning each patient to a specific room. The main differences of the proposed MIP/IP models in the literature are the hard and soft constraints in the models with respect to the original problem statement. As mentioned in Section 2.1, Demeester et al. (2010) first proposed a IP model considering the soft constraint SC1, SC2, SC3, SC4 as hard constraints and not allowing their violations. In contrast, Guido et al. (2018) proposed two IP models where HM_{PBA} does not allow the soft constraint SC1, SC2, SC3, SC4, SC9 to be violated, and SM_{PBA} relaxes the restrictions of SC1 (gender policy N,D), SC2, SC3, SC9 (patient transfer are restricted to at most one for those who have two stays). Both Ceschia & Schaerf (2011) and Bastos et al. (2019) proposed IP/MIP models which allowed all soft constraints SC1 - SC9 to be violated. However, the former is more simplified than the latter, as it merges penalties associated with the patient-room assignment, including SC1 (gender policy M, F, N), SC2, SC3, SC4, SC5, SC6, SC7, and SC8, into a single penalty C_{pr} and avoids the generation of too many constraints and variables. Moreover, it is evident that the optimal solutions, in some instances, obtained by solving the IP/MIP models of Ceschia & Schaerf (2011) and Bastos et al. (2019) outperform those of Demeester et al. (2010) and Guido et al. (2018), since the latter two models only allowed a subset of soft constraints. Based on the comparative analysis of the existing literature, it can be concluded that the IP model proposed by Ceschia & Schaerf (2011) is more flexible compared to the other models since it allows all soft constraints to be violated.

Since our work is based on the IP model proposed by Ceschia & Schaerf (2011), we summarize their reformulation below while the used notation is shown in Table 3. The objective function, denoted by (2), aims to minimize the total penalties associated with assigning patients to rooms for the duration of their hospitalization period. The first part of the objective function corresponds to the cost of assigning patients to rooms, which is determined by the combined penalty of soft constraints SC1 (gender policy M, F, N), SC2, SC3, SC4, SC5, SC6, SC7, and SC8. The second part of the objective function incorporates the cost of violating the room gender policy, while the last part of the function captures the cost associated with patient transfer.

Table 3
Notation used for the PRA model.

Symbol	Description
Sets	
$\mathcal{P}$	Set of patients ($p = 1, \cdots, \lvert\mathcal{P}\rvert$)
$\mathcal{D}$	Set of days ($d = 1, \cdots, \lvert\mathcal{D}\rvert$)
$\mathcal{R}$	Set of rooms ($r = 1, \cdots, \lvert\mathcal{R}\rvert$)
$\mathcal{D}_p \subset \mathcal{D}$	Set of days of patient p stay in hospital ($\mathcal{D}_p = \{AD_p, \cdots, DD_p - 1\}$)
$\mathcal{P}_M \subset \mathcal{P}$	Set of male patients
$\mathcal{P}_F \subset \mathcal{P}$	Set of female patients
$\mathcal{R}_D \subset \mathcal{R}$	Set of dependent rooms
Parameters	
Q_r	Number of beds in room r
LOS_p	Length of stay of patient p ($LOS_p = DD_p - AD_p$)
C_{pr}	the penalty of assigning patient p to room r. All the room penalties are incorporated into the value except SC1 (gender policy D) and SC9
W_{RG}	Weight of gender policy constraint
W_{Tr}	Weight of transfers constraint
Variables	
x_{prd}	1 if patient p is assigned to room r in day d, 0 otherwise
f_{rd}	1 if there is at least one female patient in room r in day d, 0 otherwise
m_{rd}	1 if there is at least one male patient in room r in day d, 0 otherwise
b_{rd}	1 if there are both male and female patients in room r in day d, 0 otherwise
t_{prd}	1 if patient p is transferred from room r in day d, 0 otherwise

PRA:

$$\text{Min} \sum_{p\in\mathcal{P},r\in\mathcal{R},d\in\mathcal{D}_p} C_{pr}\cdot x_{prd} + \sum_{r\in\mathcal{R}_D,d\in\mathcal{D}} W_{RG}\cdot b_{rd} + \sum_{p\in\mathcal{P},r\in\mathcal{R},d\in\mathcal{D}} W_{Tr}\cdot t_{prd} \tag{2}$$

s.t.

$$\sum_{r\in\mathcal{R}} x_{prd} = 1, \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_p \tag{3}$$

$$\sum_{p\in\mathcal{P}|d\in\mathcal{D}_p} x_{prd} \le Q_r, \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{4}$$

$$f_{rd} \ge x_{prd}, \quad \forall p \in \mathcal{P}_F, d \in \mathcal{D}, r \in R \tag{5}$$

$$m_{rd} \ge x_{prd}, \quad \forall p \in \mathcal{P}_M, d \in \mathcal{D}, r \in R \tag{6}$$

$$b_{rd} \ge m_{rd} + f_{rd} - 1, \quad \forall d \in \mathcal{D}, r \in R \tag{7}$$

$$t_{prd} \ge x_{prd} - x_{pr,d+1}, \quad \forall p \in \mathcal{P}, d \in \mathcal{D}, r \in \mathcal{R} \tag{8}$$

$$x_{prd} \in \{0,1\} \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_p, r \in \mathcal{R} \tag{9}$$

$$b_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{10}$$

$$f_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{11}$$

$$m_{rd} \in \{0,1\} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{12}$$

$$t_{prd} \in \{0,1\} \quad \forall p \in \mathcal{P}, d \in \mathcal{D}_p, r \in \mathcal{R} \tag{13}$$

Constraint (3) refers to complete assignment constraint which enforces every patient to be assigned to a room between admission and discharge dates. Constraint (4) refers to capacity constraint which ensures number of patients assigned to a room for a specific day cannot exceed the capacity of the room. Constraints (5)-(7) refer to gender policy constraints (GC_0). Variable b_{rd}, m_{rd}, f_{rd} and t_{prd} are all dependent on the different circumstances of the x_{prd} variables, which define the actual search space. If there is a female in a room, Constraint (5) forces the auxiliary variable f_{rd} to be equal to 1 to reflect the female existence in that room. A similar approach is taken for constraint (6) to reflect that there is a male in a room. If both genders exist in a room, Constraint (7) ensures that b_{rd} becomes 1 and gender penalty in the objective value is reflected accordingly. Constraint (8) refers to patient transfer constraint (TC), which ensures the auxiliary variable t_{prd} becomes 1 if a patient changes room on two consecutive days. Constraints (9) - (13) define the domain of the variables.

4 Solution approach

To solve the PAS problem, we employ a two-stage optimization approach, which decomposes the given problem into two separate subproblems: a patient-room assignment (PRA) subproblem and a patient-bed assignment (PBA) subproblem. Fig. 1 illustrates the general framework of our proposed solution approach. As demonstrated by Ceschia & Schaerf (2011), the optimal solutions derived from these two subproblems can be integrated to achieve the optimal solution for the original problem. Our approach first generates a partial solution by solving an advanced PRA (APRA) model, which is based on the IP model of Ceschia & Schaerf (2011). However, it is challenging to solve the APRA model directly because of the huge search space resulting from patient-room-day assignment variables. Thus, we employ a warm start approach in which we solve the APRA model without transfers constraint (APRA^{WT}) to generate a high-quality feasible solution and then use the obtained solution as an initial solution to the APRA model. It is worth noting that using the above warm start approach can yield better results for the tested benchmark instances than directly solving the APRA model, as demonstrated in Bastos et al. (2019). Secondly, our approach solves the PBA subproblem to allocate patients to beds of specific rooms according to the PRA solution, which is validated by an application made available online[1] by Demeester et al. (2010).

[1] https://people.cs.kuleuven.be/~wim.vancroonenburg/pas/

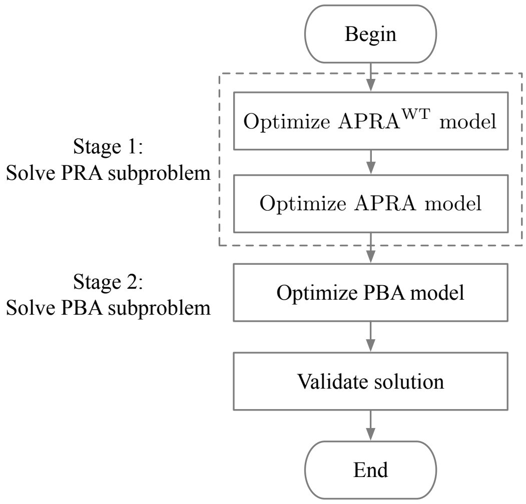


Fig. 1. Framework of two-stage optimization approach for the PAS problem.

4.1 APRA Model

Large IP models are incapable of finding a high-quality solution within an acceptable time due to their large sizes. Let the parameter G_r represent the gender policy of room r, with the values 0, 1, 2, and 3 corresponding to the policies D, M, F, and N, respectively. The variables and constraints of the IP model of Ceschia & Schaerf (2011) can be decreased by considering the following rules:

(1) Variables x_{prd} can be omitted from the model when $LOS_p = 0$.
(2) Variables f_{rd}, m_{rd} and b_{rd} can be omitted from the model when $Q_r = 1$ or $G_r = 1, 2, 3$.
(3) Variables t_{prd} can be omitted from the model when $LOS_p < 2$ and $d = DD_p - 1$.
(4) Constraints (3) and (9) can be omitted from the model when $LOS_p = 0$.
(5) Constraints (5), (6) and (7) can be omitted from the model when $LOS_p = 0$, or when either $Q_r = 1$ or $G_r = 1, 2, 3$.
(6) Constraint (8) can be omitted from the model when $Q_r = 1$ or $G_r = 1, 2, 3$.

In order to better apply the above rules, we introduce some notations presented in Table 4. It should be noted that a patient may have multiple specialisms in some instances, which means that during a patient's stay, the patient is assigned in the first part of his/her stay to a specialism, and the second part of his/her stay to another specialism. Also, the use of the parameter C_{pr} may lead to incorrect results when patients have multiple specialisms during their hospital stay. We therefore introduce a new parameter C_{prd} which is defined as the penalty of assigning patient p to room r on day d.

Table 4
Notation used for the APRA model.

Symbol	Description
Sets	
$\mathcal{P}^E \subset \mathcal{P}$	Set of elective patients with $LOS_p \geq 1$
$\mathcal{M} \subset \mathcal{P}^E$	Set of male elective patients
$\mathcal{F} \subset \mathcal{P}^E$	Set of female elective patients
$\mathcal{R}_D^M \subset \mathcal{R}$	Set of dependent rooms with more than one bed
Parameter	
C_{prd}	Penalty of assigning patient p to room r on day d. All the room penalties are incorporated into the parameter except SC1 (gender policy D) and SC9

To avoid confusion, we refer to the APRA model under the gender policy constraint GC_0 and the transfer constraint TC as $APRA_{GC_0\&TC}$, which can be formulated as follows:

$$\boldsymbol{APRA_{GC_0\&TC}}: \ \text{Min} \ \ S = \sum_{p\in\mathcal{P}^E, r\in\mathcal{R}, d\in\mathcal{D}_p} C_{prd}\cdot x_{prd} \sum_{r\in\mathcal{R}_D^M}, d\in\mathcal{D}W_{RG}\cdot b_{rd} + \sum_{p\in\mathcal{P}^E|LOS_p\geq 2, r\in\mathcal{R}, d\in\mathcal{D}_p\setminus\{DD_p-1\}} W_{Tr}\cdot t_{prd} \tag{14}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}} x_{prd} = 1, \quad \forall p\in\mathcal{P}^E, d\in\mathcal{D}_p \tag{15}$$

$$\sum_{p\in\mathcal{P}^E|d\in\mathcal{D}_p} x_{prd} \leq Q_r, \quad \forall d\in\mathcal{D}, r\in\mathcal{R} \tag{16}$$

$$f_{rd} \geq x_{prd}, \quad \forall p\in\mathcal{F}, d\in\mathcal{D}_p, r\in\mathcal{R}_D^M \tag{17}$$

$$m_{rd} \geq x_{prd}, \quad \forall p\in\mathcal{M}, d\in\mathcal{D}_p, r\in\mathcal{R}_D^M \tag{18}$$

$$b_{rd} \geq f_{rd} + m_{rd} - 1, \quad \forall d\in\mathcal{D}, r\in\mathcal{R}_D^M \tag{19}$$

$$t_{prd} \geq x_{prd} - x_{pr,d+1}, \quad \forall p\in\mathcal{P}^E|LOS_p\geq 2, d\in\mathcal{D}_p\setminus\{DD_p-1\}, r\in\mathcal{R} \tag{20}$$

$$x_{prd}\in\{0,1\} \quad \forall p\in\mathcal{P}^E, d\in\mathcal{D}_p, r\in\mathcal{R} \tag{21}$$

$$b_{rd}\in\{0,1\} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}_D^M \tag{22}$$

$$f_{rd}\in\{0,1\} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}_D^M \tag{23}$$

$$m_{rd}\in\{0,1\} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}_D^M \tag{24}$$

$$t_{prd}\in\{0,1\} \quad \forall p\in\mathcal{P}^E|LOS_p\geq 2, d\in\mathcal{D}_p\setminus\{DD_p-1\}, r\in\mathcal{R} \tag{25}$$

Regarding gender policy constraints, we can modify the formulas presented in Bastos et al. (2019) to obtain alternative formulations (GC_1). We define a new binary variable u_{rd}, which has the value 1 if room r is reserved for females on day d, and 0 otherwise. The constraints GC_1 can be written as follows:

$$(GC_1) \qquad x_{prd} \leq u_{rd} + b_{rd} \quad \forall p\in\mathcal{F}, d\in\mathcal{D}_p, r\in\mathcal{R}_D^M \tag{26}$$

$$x_{prd} \leq (1-u_{rd}) + b_{rd} \quad \forall p\in\mathcal{M}, d\in\mathcal{D}_p, r\in\mathcal{R}_D^M \tag{27}$$

$$u_{rd}\in\{0,1\} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}_D^M \tag{28}$$

Constraint (26) enforces female patient restrictions and (27) enforces male patient restrictions. Both constraints seek to avoid the assignment of two distinct genders to the same room, penalizing allocations in which different genders share a room. Thus, we refer the APRA model under constraints GC_1 and TC as $APRA_{GC_1\&TC}$.

4.2 APRA model without transfers constraint

It is quite challenging to solve the above two APRA models directly due to the large search space defined by the patient-room-day assignment variables. By prohibiting patient transfer during their stay, we limit the scope of the search space defined by patient-room assignment variables, resulting in a special case of the APRA model, as used in Bastos et al. (2019); Guido et al. (2018); Ceschia & Schaerf (2011). In our APRA$^{\text{WT}}$ model, transfers are not allowed so that a patient must stay in the same room during his/her entire length of stay. The solution of APRA$^{\text{WT}}$ model will always be feasible to the APRA model since the solution space of the former is contained in that of the latter. Moreover, since transfers are associated with the highest penalty weight, it is reasonable to expect that the solution of APRA$^{\text{WT}}$ models would be close to the optimal solution of APRA models. Hence, we first solve an APRA$^{\text{WT}}$ model to obtain a feasible solution, which is used as the initial solution of the APRA model for further improvement.

Our APRA$^{\text{WT}}$ models are inherited from the APRA models by removing variable t_{prd} and replacing variable x_{prd} by x_{pr}, a binary variable taking the value of 1 if patient p is allocated to room r, and 0 otherwise. Moreover, parameter C_{prd} is replaced by parameter $C'_{pr} = \sum_{d\in\mathcal{D}_p} C_{prd}$. Thus, the model $APRA^{WT}_{GC_0}$ is formulated as follows:

$$\boldsymbol{APRA^{WT}_{GC_0}}: \quad \text{Min} \quad S = \sum_{p\in\mathcal{P}^E}\sum_{r\in\mathcal{R}} C'_{pr}x_{pr} + \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}^M_D} W_{RG}b_{rd} \tag{29}$$

s.t. constraints (19), (22), (23)

$$\sum_{r\in\mathcal{R}} x_{pr} = 1 \quad \forall p\in\mathcal{P}^E \tag{30}$$

$$\sum_{p\in\mathcal{P}^E|d\in\mathcal{D}_p} x_{pr} \le Q_r \quad \forall d\in\mathcal{D}, r\in\mathcal{R} \tag{31}$$

$$f_{rd} \ge x_{pr} \quad \forall p\in\mathcal{F}, d\in\mathcal{D}_p, r\in\mathcal{R}^M_D \tag{32}$$

$$m_{rd} \ge x_{pr} \quad \forall p\in\mathcal{M}, d\in\mathcal{D}_p, r\in\mathcal{R}^M_D \tag{33}$$

$$x_{pr} \in \{0,1\} \quad \forall p\in\mathcal{P}^E, r\in\mathcal{R} \tag{34}$$

Like the APRA model, constraint (30) refers to complete assignment constraint, constraint (31) refers to capacity constraint and constraints (19),(32)-(33) refer to gender policy constraints (GC_0). Furthermore, model $APRA^{WT}_{GC_1}$

can be obtained by replacing GC_0 with GC_1 (28), (35)-(36) in model $APRA^{WT}_{GC_0}$.

$$(GC_1) \qquad \text{Constraint (28)}$$

$$x_{pr} \leq u_{rd} + b_{rd} \quad \forall p \in \mathcal{F}, d \in \mathcal{D}_p, r \in \mathcal{R}_D^M \tag{35}$$

$$x_{pr} \leq (1 - u_{rd}) + b_{rd} \quad \forall p \in \mathcal{M}, d \in \mathcal{D}_p, r \in \mathcal{R}_D^M \tag{36}$$

4.3 Constraint aggregation

Despite using the rules we proposed in Section 4.1 to reduce the model size, the APRA and APRA$^{\text{WT}}$ models are still large and hard to solve. To further accelerate the solution process, we propose constraint aggregation to reduce the number of gender policy constraints GC_0, GC_1 and patient transfer constraint TC considering that these constraints account for more than 95% of the total number of constraints in the APRA and APRA$^{\text{WT}}$ models (see Section 5.1). Since the APRA$^{\text{WT}}$ models are inherited from the APRA models, we take the APRA models as example to illustrate our aggregation method.

4.3.1 Aggregated gender policy constraint

For gender policy constraints GC_1, we propose aggregated gender policy constraints AGC_1 (28), (37)-(38) by aggregating the constraints (26)-(27) of different patients with the same gender for the same day and room.

$$(AGC_1) \qquad \text{Constraint (28)}$$

$$\sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd} \leq \lambda^F_{rd}(u_{rd} + b_{rd}) \quad \forall r \in \mathcal{R}_D^M, d \in \mathcal{D} \tag{37}$$

$$\sum_{p \in \mathcal{M} | d \in \mathcal{D}_p} x_{prd} \leq \lambda^M_{rd}(1 - u_{rd} + b_{rd}) \quad \forall r \in \mathcal{R}_D^M, d \in \mathcal{D} \tag{38}$$

where $\lambda^F_{rd} = \min\{Q_r, |\mathcal{F}_d|\}$ and $\lambda^M_{rd} = \min\{Q_r, |\mathcal{M}_d|\}$ are coefficients, $|\mathcal{F}_d|$ and $|\mathcal{M}_d|$ are the number of female/male elective patients in day d. Thus, the aggregated model $APRA_{AGC_1\&TC}$ can be obtained by replacing GC_1 with AGC_1 in model $APRA_{GC_1\&TC}$.

The $APRA_{GC_1\&TC}$ is equivalent to the $APRA_{AGC_1\&TC}$ (see the proof of Theorem 1) under the following two conditions: (i) if ($\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{b}$, $\boldsymbol{t}$) is a feasible solution to the $APRA_{GC_1\&TC}$, then it must be feasible to the $APRA_{AGC_1\&TC}$; (ii) if ($\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{b}$, $\boldsymbol{t}$) is a feasible solution to the $APRA_{AGC_1\&TC}$, then it must be feasible to the $APRA_{GC_1\&TC}$.

Theorem 1 *The aggregated model $APRA_{AGC_1\&TC}$ is equivalent to the original model $APRA_{GC_1\&TC}$.*

Proof. The difference between the models $APRA_{AGC_1\&TC}$ and $APRA_{GC_1\&TC}$ is the constraints (26)-(27) and (37)-(38). Thus, the other constraints are omitted in the following proof. It is worth noting that the variables of (26) and (27) are consistent, and the structure of these two constraints are similar, the method used to prove the equivalence between (26) and (37) can be applied to prove the equivalence between (27) and (38). Consequently, we will only present the proof of the equivalence of constraints (26) and (37).

Proof of condition (i): By summing up the inequalities indexed by p in constraint (26), we get inequality (39).

$$|\mathcal{F}_d|(b_{rd} + u_{rd}) \geq \sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd} \quad \forall r \in \mathcal{R}_D^M, d \in \mathcal{D} \tag{39}$$

According to the domains of the variables b_{rd}, u_{rd}, x_{prd}, we consider the following two cases for the inequality (39):
Case 1: If $b_{rd} + u_{rd} = 0$, then $\sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd} = 0$, which implies that constraint (37) is satisfied.
Case 2: If $b_{rd} + u_{rd} > 0$, then we can rewrite (39) as:

$$\frac{\sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd}}{b_{rd} + u_{rd}} \leq |\mathcal{F}_d| \quad \forall r \in \mathcal{R}_D^M, d \in \mathcal{D} \tag{40}$$

Since the number of female elective patients assgined to room r on day d is no more than the capacity of room r, and the number of female elective patients on day d, i.e., $\sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd} \leq \min\{Q_r, |\mathcal{F}_d|\} = \lambda_{rd}^F, \forall r \in \mathcal{R}_D^M, d \in \mathcal{D}$, it follows that the left-hand side of (40) is no more than λ_{rd}^F. As a result, constraint (37) is also satisfied in this case. Thus, condition (i) is proved.

Proof of condition (ii): By rewriting the constraint (37), we have:

$$b_{rd} + u_{rd} \geq \frac{\sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd}}{\lambda_{rd}^F} \geq x_{prd} \quad \forall p \in \mathcal{F}, d \in \mathcal{D}_p, r \in \mathcal{R}_D^M \tag{41}$$

Obviously, the constraint (26) is satisfied. Thus, we have proved condition (ii). Therefore, we have proved the equivalence of the two models. □

The gender policy constraints GC_0 (17)-(19) can be reformulated by AGC_0 (19), (42)-(43) as follows:

(AGC_0) Constraint (19)

$$\lambda_{rd}^F f_{rd} \geq \sum_{p \in \mathcal{F} | d \in \mathcal{D}_p} x_{prd} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_D^M \tag{42}$$

$$\lambda_{rd}^M m_{rd} \geq \sum_{p \in \mathcal{M} | d \in \mathcal{D}_p} x_{prd} \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_D^M \tag{43}$$

Thus, the aggregated model $APRA_{AGC_0\&TC}$ can be obtained by replacing GC_0 with AGC_0 in model $APRA_{GC_0\&TC}$. With reference to the proof of Theorem

1, it is easy to see that the aggregated model $APRA_{AGC_0\&TC}$ is equivalent to the original model $APRA_{GC_0\&TC}$.

To obtain the aggregated gender policy constraints used in the APRA$^{\text{WT}}$ models, we can replace the variable $x_{prd}, \forall p \in \mathcal{P}^E, d \in \mathcal{D}_p, r \in \mathcal{R}$ with $x_{pr}, \forall p \in \mathcal{P}^E, r \in \mathcal{R}$. This allows us to create two APRA$^{\text{WT}}$ models, denoted as $APRA^{WT}_{AGC_0}$ and $APRA^{WT}_{AGC_1}$. Additionally, the aggregated models $APRA^{WT}_{AGC_0}$ and $APRA^{WT}_{AGC_1}$ are equivalent to the original models $APRA^{WT}_{GC_0}$ and $APRA^{WT}_{GC_1}$, respectively.

4.3.2 Aggregated patient transfer constraint

To aggregate patient transfer constraint, variable $t_{prd}, \forall p \in \mathcal{P}^E, d \in \mathcal{D}_p, r \in \mathcal{R}$ is replaced by aggregated variable $z_{pd}, \forall p \in \mathcal{P}^E, d \in \mathcal{D}_p$, a binary variable taking the value of 1 if patient p is transferred to a new room in day d, and 0 otherwise. With this new aggregated variable, the objective function (4.1) need to be modified by (44).

$$\begin{aligned}\text{Min} \quad S = &\sum_{p\in\mathcal{P}^E, r\in\mathcal{R}, d\in\mathcal{D}_p} C_{prd}\cdot x_{prd} + \sum_{r\in\mathcal{R}^M_D}, d\in\mathcal{D}W_{RG}\cdot b_{rd} + \\ &\sum_{p\in\mathcal{P}^E|LOS_p\geq 2, d\in\mathcal{D}_p\setminus\{DD_p-1\}} W_{Tr}\cdot z_{pd}\end{aligned} \tag{44}$$

Aggregate patient transfer constraint can be achieved by comparing the room number RN_r of two consecutive days to determine whether the patient is transferred. Therefore, aggregated patient transfer constraint ATC (45) - (47) can be reformulated as follows:

$$(ATC) \quad |\mathcal{R}|z_{pd} \geq \sum_{r\in\mathcal{R}} RN_r x_{prd} - \sum_{r\in\mathcal{R}} RN_r x_{pr,d+1}$$
$$\forall p\in\mathcal{P}^E|LOS_p\geq 2, d\in\mathcal{D}_p\setminus\{DD_p-1\} \tag{45}$$
$$|\mathcal{R}|z_{pd} \geq \sum_{r\in\mathcal{R}} RN_r x_{pr,d+1} - \sum_{r\in\mathcal{R}} RN_r x_{prd}$$
$$\forall p\in\mathcal{P}^E|LOS_p\geq 2, d\in\mathcal{D}_p\setminus\{DD_p-1\} \tag{46}$$
$$z_{pd}\in\{0,1\} \quad \forall p\in\mathcal{P}^E|LOS_p\geq 2, d\in\mathcal{D}_p\setminus\{DD_p-1\} \tag{47}$$

Four APRA models can be generated by combining the objective function (44), the complete assignment constraint, the capacity constraint, along with different formulations of the (aggregated) gender policy constraint and the aggregated patient transfer constraint, namely $APRA_{GC_0\&ATC}$, $APRA_{GC_1\&ATC}$, $APRA_{AGC_0\&ATC}$, $APRA_{AGC_1\&ATC}$. Furthermore, each of these aggregated APRA models is equivalent to its corresponding APRA model under the constraint TC. For instance, $APRA_{GC_0\&ATC}$ is equivalent to $APRA_{GC_0\&TC}$, $APRA_{GC_1\&ATC}$ is equivalent to $APRA_{GC_1\&TC}$, $APRA_{AGC_0\&ATC}$ is equivalent to $APRA_{AGC_0\&TC}$, $APRA_{AGC_1\&ATC}$ is equivalent to $APRA_{AGC_1\&TC}$. See the proof of Theorem 2 for details.

Theorem 2 *The APRA model under the aggregated constraint ATC is equivalent to its corresponding APRA model under the constraint TC.*

Proof. Given two APRA models, i.e. $APRA_{GC_1\&TC}$ and $APRA_{GC_1\&ATC}$. In order to prove the equivalence of the two models, we need to prove the following two conditions: (i) if $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$ is a feasible solution to the $APRA_{GC_1\&TC}$, then, there exists a vector $\boldsymbol{z}$ such that $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z})$ is feasible to the $APRA_{GC_1\&ATC}$ with objective value $S_{APRA_{GC_1\&ATC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z}) = S_{APRA_{GC_1\&TC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$. (ii) if $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z})$ is a feasible solution to the $APRA_{GC_1\&ATC}$, then, there exists a vector $\boldsymbol{t}$ such that $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$ is feasible to the $APRA_{GC_1\&TC}$ with objective value $S_{APRA_{GC_1\&TC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t}) = S_{APRA_{GC_1\&ATC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z})$. The difference between the two models is the objective function and the constraints TC and ATC. Thus, the other constraints are omitted in the following proof. For $\forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}$, let $a(p, d)$ be the room index r where $x_{prd} = 1$.

Proof of condition (i): Since complete assignment constraint (15) holds, the right-hand side of (45) gives $RN_{a(p,d)} - RN_{a(p,d+1)} \in [1 - |R|, |R| - 1]$ and the right-hand side of (46) gives $RN_{a(p,d+1)} - RN_{a(p,d)} \in [1 - |R|, |R| - 1]$. If $a(p, d) \neq a(p, d + 1)$, then $z_{pd} = 1$. If $a(p, d) = a(p, d + 1)$, then both $z_{pd} = 0$ and $z_{pd} = 1$ can satisfy (45) and (46). Hence, $\boldsymbol{z}$ can be determined. $S_{APRA_{GC_0\&ATC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{z}) = S_{APRA_{GC_1\&TC}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{t})$ is proved by observing that each patient can only change room on two consecutive days once. Thus, $\sum_{r \in \mathcal{R}} t_{prd} = z_{pd}$, $\forall p \in \mathcal{P}^E | LOS_p \geq 2, d \in \mathcal{D}_p \setminus \{DD_p - 1\}$. Therefore, the objective of the two models are the same. This proves the condition (i).

Proof of condition (ii): We discuss the proof of condition (ii) by considering two cases. Case 1: $a(p, d) = a(p, d + 1)$. In this case, based on the domain of the variable x_{prd}, the right-hand side of (20) gives 0. Therefore, both $t_{prd} = 0$ and $t_{prd} = 1$ can satisfy the constraint (20). Case 2: $a(p, d) \neq a(p, d + 1)$. In this case, $t_{p,a(p,d),d} = 1$, and for $\forall r \in \mathcal{R} \backslash \{a(p, d)\}$, both $t_{prd} = 0$ and $t_{prd} = 1$ can satisfy the constraint (20). Hence, $\boldsymbol{t}$ can be determined. Similar to the proof of condition (i), the two models have the same the objectives. Thus, we have proved condition (ii).

Therefore, we have proved the equivalence of the two models. □

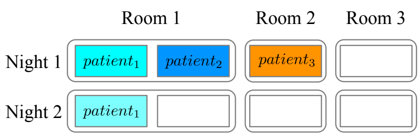
4.3.3 An illustrative example

To illustrate our proposed aggregation method, consider an illustrative example of the PRA subproblem with 3 elective patients, 3 rooms, and 2 nights, as shown in Figure 2. Room 1, with 2 beds, follows policy D. Rooms 2 and 3 are both 1-bed rooms and follow policy M and F, respectively. The table on the top left lists the input data related to the patients, where the meaning of the symbols corresponds to the definition in Tables 3 and 4. The lower part of the table lists the constraints related to gender policy and patient transfer. The gender policy constraint GC_0 consists of 6 inequalities, whereas its aggregated counterpart AGC_0 contains 5 inequalities. Similarly, GC_1 involves 4 inequalities, while AGC_1 contains 3. For patient transfer, TC has 3 inequalities, while ATC has 2 inequalities. All 8 APRA models can be obtained by combining different formulations of the (aggregated) gender policy constraint and the (aggregated)patient transfer constraint, as well as the objective function, the complete assignment constraint, and the capacity constraint (here, the objective function, the complete assignment constraint, and the capacity constraint are omitted for brevity). All these APRA models have the same optimal solution, which is shown in the figure on the top right. The optimal objective function value is $S = 13.6$.

Instance

D_p	LOS_p	C_{prd}		
		$r=1$	$r=2$	$r=3$

Optimal Solution: $S = 13.6$



(Aggregated) gender policy constraints and (aggregated) patient transfer constraints

Original	GC_0 $f_{1,1} \geq x_{1,1,1}$ $f_{1,2} \geq x_{1,1,2}$ $f_{1,1} \geq x_{2,1,1}$ $m_{1,1} \geq x_{3,1,1}$ $b_{1,1} \geq f_{1,1} + m_{1,1} - 1$ $b_{1,2} \geq f_{1,2} + m_{1,2} - 1$	GC_1 $x_{1,1,1} \leq u_{1,1} + b_{1,1}$ $x_{2,1,1} \leq u_{1,1} + b_{1,1}$ $x_{1,1,2} \leq u_{1,2} + b_{1,2}$ $x_{3,1,1} \leq 1 - u_{1,1} + b_{1,1}$	TC $t_{1,1,1} \geq x_{1,1,1} - x_{1,1,2}$ $t_{1,2,1} \geq x_{1,2,1} - x_{1,2,2}$ $t_{1,3,1} \geq x_{1,3,1} - x_{1,3,2}$
Aggregation	AGC_0 $2f_{1,1} \geq x_{1,1,1} + x_{2,1,1}$ $f_{1,2} \geq x_{1,1,2}$ $m_{1,1} \geq x_{3,1,1}$ $b_{1,1} \geq f_{1,1} + m_{1,1} - 1$ $b_{1,2} \geq f_{1,2} + m_{1,2} - 1$	AGC_1 $x_{1,1,1} + x_{2,1,1} \leq 2(u_{1,1} + b_{1,1})$ $x_{1,1,2} \leq u_{1,2} + b_{1,2}$ $x_{3,1,1} \leq 1 - u_{1,1} + b_{1,1}$	ATC $3z_{1,1} \geq x_{1,1,1} + 2x_{1,2,1} + 3x_{1,3,1} - x_{1,1,2} - 2x_{1,2,2} - 3x_{1,3,2}$ $3z_{1,1} \geq x_{1,1,2} + 2x_{1,2,2} + 3x_{1,3,2} - x_{1,1,1} - 2x_{1,2,1} - 3x_{1,3,1}$

Fig. 2. An illustrative example of PRA subproblem

It is worth noting that using the above aggregated constraint AGC_0, AGC_1 and ATC can reduce the number of constraints of APRA models, but it may also bring some disadvantages when solving the aggregated models using the Branch-and-Bound (B&B) approach. The B&B approach uses a lower bounding strategy based on linear program (LP) relaxation. However, aggregation enlarges the set of feasible solutions of the LP relaxation, thereby leading to weaker lower bounds. The weakened lower bounds may reduce the effectiveness of the B&B approach. The related detailed discussion can be found in Khurana & Murty (2012), where the authors studied the effects of aggregation on the computational ease of the model. Therefore, to verify the effectiveness of the proposed aggregated constraints, we will compare the computational results of the models with and without aggregated constraints in Section 5.1.

4.4 PBA model

The PBA subproblem is created based on the outputs of the PRA subproblem to generate the patient-bed assignments. To build the PBA model, the hospital stay segment is introduced to indicate the patient transfer. If the room assigned to patient p in day d is the same as the room in consecutive hospitalization days, these days belong to the same hospital stay segment s. The sets of hospital stay segments for all patients can be easily calculated according to the results of the PRA subproblem. The PBA is a feasibility problem, which is solved with a constant objective function set to zero, as shown in (48). The notation for the PBA is provided in Table 5, and its formulation is as follows:

Table 5
PBA Notation

Symbol	Description
Sets	
$\mathcal{B}$	Set of beds ($b = 1, \cdots, \lvert\mathcal{B}\rvert$)
$\mathcal{S}_p$	Set of hospital stay segements of patient p ($s = 1, \cdots, \lvert\mathcal{S}_p\rvert$)
$\mathcal{B}_r \subset \mathcal{B}$	Set of beds in room r
$\mathcal{P}_{rd} \subset \mathcal{P}$	Set of patients assigned to room r in day d
$\mathcal{D}_{ps} \subset \mathcal{D}$	Set of days in hospital stay segement s of patient p
Parameters	
G_{ps}	Room assigned to patient p in his/her hospital stay segment s
Variables	
y_{pbs}	1 if patient p is assigned to bed b in hospital stay segment s, 0 otherwise

$$\text{Min } 0 \tag{48}$$

$$\text{s.t.} \quad \sum_{b \in \mathcal{B}_{G_{ps}}} y_{pbs} = 1 \quad \forall p \in \mathcal{P}, s \in \mathcal{S}_p \tag{49}$$

$$\sum_{p \in \mathcal{P}_{rd}} \sum_{s \in \mathcal{S}_p | d \in \mathcal{D}_{ps}, r = G_{ps}} y_{pbs} \leq 1 \quad \forall r \in \mathcal{R}, b \in \mathcal{B}_r, d \in \mathcal{D} \tag{50}$$

$$y_{pbs} \in \{0, 1\} \quad p \in \mathcal{P}, s \in \mathcal{S}_p, b \in \mathcal{B}_{G_{ps}} \tag{51}$$

Constraint (49) ensures every patient to be assigned to a bed for each segment. Constraint (50) limits assignments to the capacity of each bed for each night. Constraint (51) define the domain of the decision variable.

5 Results and comparisons

In this section, we present computational results of our proposed solution method on the 13 instances provided by Demeester et al. (2008). Table 6 shows the details of these 13 instances in terms of the number of rooms ($|\mathcal{R}|$), dependent rooms with more than one bed ($|\mathcal{R}_D^M|$), total patient ($|\mathcal{P}|$), elective patient ($|\mathcal{P}^E|$), room properties (*Prop.*), beds ($|\mathcal{B}|$), specialisms (S), the length of the planning horizon ($|\mathcal{D}|$), and departments (K). In addition, only three room sizes are considered in this benchmark, i.e. 1, 2 and 4 beds.

The first six instances benefit from better patient-room compatibility compared with instances 7-13. In addition, the planning horizon is 14 days for instances 1-7, which is smaller than instances 8-13, where the planning horizon is between 21 days to 91 days. Therefore, instances 8-13 are more complex. It is worth mentioning that the total number of patients includes elective patients as well as patients whose LOS is zero, and patients whose discharge date lies beyond the planning horizon are scheduled until the last planning day. Moreover, only instance 13 has multi-spec patients. Specifically, all the 202 multi-spec patients require two specialisms, and no patient requires more than two specialisms in this benchmark dataset.

Our model was implemented and solved using Gurobi Optimizer 9.0.3 with its default parameter settings. Branch-and-cut (B&C) is the default algorithm of Gurobi to solve the MIP models. Experiments are run on a cluster with each node running Linux with Inter(R) Xeon(R) Gold 6226R 2.90GHz CPU and 256Gb RAM. The number of CPU cores used was set to be 10. Experiments revealed that the average time to generate patient-room penalty matrix takes no more than 10 seconds, and solving the PBA model takes no more than 1 second. Thus, given a total time limit, we set 50 % of the run time to solve the APRA$^{\mathrm{WT}}$ model and set the remaining time to solve the PRA model, which is the same as Bastos et al. (2019).

Table 6
Characteristics of the problem instances.

Instance	$\|\mathcal{R}\|$	$\|\mathcal{R}_D^M\|$	$\|\mathcal{P}\|$	$\|\mathcal{P}^E\|$	*Prop.*	$\|\mathcal{B}\|$	S	$\|D\|$	K
1	98	82	693	652	2	286	4	14	4
2	151	132	778	755	2	465	6	14	6
3	131	114	757	708	2	395	5	14	5
4	155	136	782	746	2	471	6	14	6
5	102	93	631	587	2	325	4	14	4
6	104	93	726	685	2	313	4	14	4
7	162	32	770	519	4	472	6	14	6
8	148	34	895	895	4	441	6	21	6
9	105	18	1400	1400	4	310	4	28	4
10	104	20	1575	1575	4	308	4	56	4
11	107	21	2514	2514	4	318	4	91	4
12	105	28	2750	2750	4	310	4	84	4
13	125	30	907	907	4	368	5	28	5

We generate 8 APRA models as well as 4 APRA$^{\mathrm{WT}}$ models, aiming to answer two critical questions: (i) do different models perform differently? (ii) if yes, which model performs the best for solving the PRA subproblem and why? In the following, we assess which model best suits the PRA subproblem using the benchmark sets. As described in Section 4, we use a warm start approach to solve the PRA subproblem, in which an APRA$^{\mathrm{WT}}$ model is solved in the initial step and then an APRA model is solved in the subsequent step. Specifically, the type of formula used for the gender policy constraint remains consistent between the APRA$^{\mathrm{WT}}$ model and the APRA model. As a result, this yields 8 different warm start procedures.

Model size can be used to roughly infer the performance of the solution. In general, a smaller model (with less constraints and variables) would be easier to handle. Thus, we first compare the average number of constraints and variables of all APRA$^{\mathrm{WT}}$/APRA models we proposed as well as MIP models of Bastos et al. (2019), which generated most of the best known solutions and lower bounds, for 13 benchmark instances, as shown in Table 7. Notice that Bastos et al. (2019) also used the warm start approach to solve the PRA subproblem, and refereed to the model used in the initial step as the *simplified model (SM)* and the model used in the subsequent step as *complete model (CM)*. Additionally, the *SM* is a special case of the *CM* forbidding patient transfer.

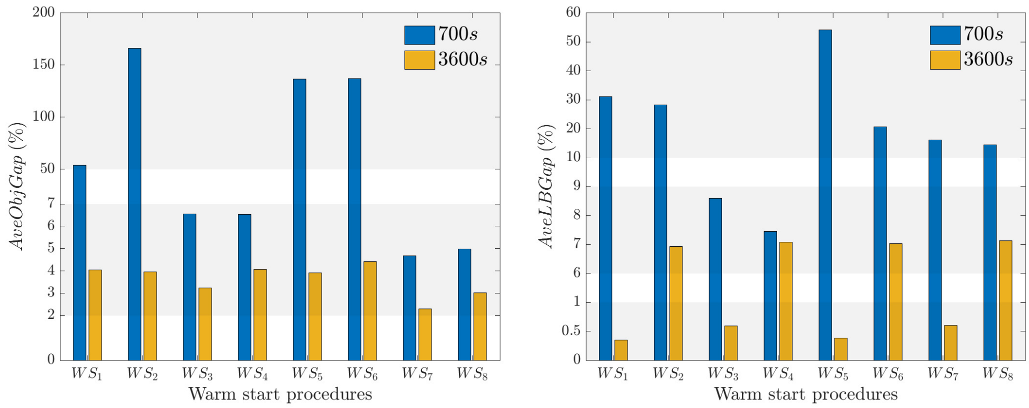
From Table 7, firstly, we can observe that our proposed models are significantly smaller than the MIP models of Bastos et al. (2019) in both initial and subsequent steps. Secondly, the model sizes of our proposed models are also significantly different. Specifically, in the initial step, the average number of variables and constraints of $APRA^{WT}_{GC_1}$/ $APRA^{WT}_{AGC_1}$ decreases by 10^3 compared to $APRA^{WT}_{GC_0}$/ $APRA^{WT}_{AGC_0}$. It is also clear that using the aggregated gender policy constraints can significantly decrease the model size. Compared to $APRA^{WT}_{GC_0}$/ $APRA^{WT}_{GC_1}$, the average number of constraints of $APRA^{WT}_{AGC_0}$/ $APRA^{WT}_{AGC_1}$ decreases by 97% after aggregating the gender policy constraints. Additionally, in the subsequent step, we can observe that the average number of constraints decreases by 33% after using $AGC_0(AGC_1)$ and the average number of constraints decreases by 65% after using ATC in APRA. If both aggregated constraints $AGC_0(AGC_1)$ and ATC are used, the average number of constraints decreases by 97%. Moreover, the average number of variables decreases 44% after using ATC in APRA.

Next, we present a comparative results by using the above warm start procedures to solve the PRA subproblem. Due to the challenge of the PAS problem, we are more concerned with the *solution-quality* than *proven-optimality*. In order to assess the performance of the above warm start procedures under dif-

Table 7
Comparison of different models used in warm start procedures over all benchmark instances

Warm start procedure	Initial step			Subsequent step		
	Model	Var.	Con.	Model	Var.	Con.
Bastos et al. (2019)						
WS_0	*SM*	5.20×10^6	6.60×10^6	*CM*	5.70×10^6	7.00×10^6
This paper						
WS_1	$APRA^{WT}_{GC_0}$	1.41×10^5	2.65×10^5	$APRA_{GC_0\&TC}$	1.27×10^6	7.87×10^5
WS_2				$APRA_{GC_0\&ATC}$	6.58×10^5	2.78×10^5
WS_3	$APRA^{WT}_{AGC_0}$	1.41×10^5	8.65×10^3	$APRA_{AGC_0\&TC}$	1.27×10^6	5.30×10^5
WS_4				$APRA_{AGC_0\&ATC}$	6.58×10^5	2.19×10^4
WS_5	$APRA^{WT}_{GC_1}$	1.40×10^5	2.64×10^5	$APRA_{GC_1\&TC}$	1.17×10^6	7.85×10^5
WS_6				$APRA_{GC_1\&ATC}$	6.57×10^5	2.77×10^5
WS_7	$APRA^{WT}_{AGC_1}$	1.40×10^5	7.35×10^3	$APRA_{AGC_1\&TC}$	1.17×10^6	5.29×10^5
WS_8				$APRA_{AGC_1\&ATC}$	6.57×10^5	2.06×10^4

ferent solution time limits, two sets of results for each constraint combination were generated. The first was obtained with a short run time limit of 700 seconds. The second was obtained with a long run time limit of 3600 seconds (or until an optimal solution is found). For each warm start procedure, we summarize the average percentage gap $AveObjGap(\%) = \sum_{i\in N} \frac{Obj_i - BKS_i}{BKS_i * N} * 100$ of the best objective values Obj obtained by our approach from the best known objective values BKS reported in (Bastos et al., 2019; Guido et al., 2018) over the N benchmark instances ($N = 13$ in our case) and illustrate the results in Figure 3(a). Similarly, the average percentage gap $AveLBGap(\%) = \sum_{i\in N} \frac{BLB_i - LB_i}{BLB_i * N} * 100$ of the best lower bounds LB from the best known lower bounds BLB (Bastos et al., 2019) are summarized in Figure 3(b). To make the results more readable, we properly scaled the vertical axis of the figures. From Figure 3(a), we observe that under the short and long run times, the



(a) Solution quality (b) Lower bound (LB) quality

Fig. 3. Performance of different models in solving the PRA subproblem

model under the aggregated gender policy constraint AGC_0 and AGC_1 (WS_3, WS_4, WS_7, WS_8) can significantly improve the solution quality, while using the aggregated patient transfer constraint ATC (WS_2, WS_6) fails to improve the solution quality. Furthermore, the models under constraint GC_0 (WS_1, WS_2) perform similarly to the models under constraint GC_1 (WS_5, WS_6), whereas the models under constraints AGC_1 always perform better than the models under AGC_0.

From Fig. 3(b), we observe that the more aggregation constraints are used, the better LBs are generated in a short run time. On the contrary, the more aggregation constraints are used, the worse LBs are generated in a long run time. Specifically, the LB quality generated by warm start procedures in a long run time are as follows: WS_1 (0.35%) $<$ WS_5 (0.38%) $<$ WS_3 (0.60%) $=$ WS_7 (0.60%) $<$ WS_2 (6.93%) $<$ WS_6 (7.03%) $<$ WS_4 (7.09%) $<$ WS_8 (7.14%). Moreover, the LBs generated by solving the models under the constraint ATC (WS_2, WS_4, WS_6, WS_8) are difficult to improve for a longer run time, while LBs generated by solving the models under the constraint TC (WS_1, WS_3, WS_5, WS_7) are easy to improve in comparison.

These results indicate that the impact of using the aggregated constraints is two-fold. First, using constraint aggregation can significantly reduce the model size and using appropriate aggregated method can make the model easier to solve. Second, solving the model under aggregated constraints may quickly generate a lower bound for the minimization problem, but the quality of the lower bound is difficult to improve in a long run time. The reason for this phenomenon can be explained as Section 4.3. In summary, this experiment demonstrates that 1) appropriate formulas of gender policy constraints are essential for solving the APRA$^{\mathrm{WT}}$ model, and 2) the APRA model as the core component of the first stage of our two-stage optimization method and $APRA_{AGC_1\&TC}$ (WS_7) using constraints AGC_1 and TC perform the best among all the models we examined.

5.2 *Comparison with state-of-the-art results*

In order to compare our best results with those obtained in previous works, we perform additional experiments by running warm start procedure WS_7 for a time limit of 24 hours as the previous works (Ceschia & Schaerf, 2011; Bastos et al., 2019). Due to the differences in computers and Gurobi versions between our work and previous studies, we implement the MIP models of Bastos et al. (2019) with warm start WS_0 to solve the PRA subproblem and use Gurobi Optimizer 9.0.3 with its default settings to solve the model.

To compare our results with previous studies, we adjusted the reported run times to account for CPU performance. We followed the approach used in Bastos et al. (2019), which is based on the approach proposed in Da Silva et al.

(2012). The performance ratings of different CPUs have been obtained online [1] and are shown in the table 8. In our study, we used 10 of 16 available cores (20 of 32 threads) on the Intel Xeon Gold 6226R 2.90 gigahertz processor. To the best of our knowledge, no public performance data exists for this specific configuration. Given that CPU performance does not scale linearly with the number of cores, we first calculated the ratio of the performance degradation as follows: $\frac{\text{Average CPU mark}}{\text{Total threads}\times\text{Single thread rating}} = \frac{26240}{2294\times 32} \approx 0.357$. Then, we calculate the estimated CPU mark for 10 cores by 20 threads $\times$ single thread rating $\times$ ratio of the performance degradation, i.e., $20 \times 2294 \times 0.357 \approx 16379$. The above approach was used to adjust the run times reported in Bastos et al. (2019) and Guido et al. (2018).

Table 9 contrasts the best known results in the literature with our best results. Under the header "Literature Results", we present the best known objective values BKS and best known lower bounds BLB for each of the 13 benchmark instances. Moreover, the reference papers and computational times (adjusted following the procedure detailed previously) associated with these values have been reported. We show the results generated by the two-stage approach with the literature's MIP model under the header "Warm start WS_0 (MIP models of Bastos et al. (2019))", and report our results generated by the two-stage approach using the best model we proposed under the header "Warm start WS_7". For each approach, we record the best objective (Obj), the total computation time to find the best solution, the total computation time when Gurobi either proves the optimality or reaches the time limit (24 hours), the best lower bound (LB) and the number of branch-and-bound nodes visited after the root node in SM, CM, APRA$^{\text{WT}}$ and APRA models. We compute the percentage gaps $GAP(\%) = \frac{Obj - LB^*}{LB^*} \times 100$ of the best objective value found by each approach from the best lower bound LB^*, which is the maximum value among the lower bounds reported in the literature as well as those obtained in our study. Furthermore, we present the objective values, total computation time and the lower bounds as reported by Bastos et al. (2019), which were obtained using the warm start approach.

Table 8
Optimization solvers and performance evaluation of CPU

Reference	Solver	Processor	Single thread rating	Average CPU mark	Used cores/ Total cores	Used threads/ Total threads
Guido et al. (2018)	Cplex 15.5.1	Intel Xeon E5-1620 3.60 gigahertz 32 gigabytes RAM	1774	5863	4*/4	8*/8
Bastos et al. (2019)	Gurobi 7.5	Intel i7-3960 × 3.3 gigahertz 64 gigabytes RAM	1793	8390	6*/6	12*/12
This paper	Gurobi 9.0.3	Intel Xeon Gold 6226R 2.90 gigahertz 256 gigabytes RAM	2294	26240 (16379**)	10/16	20/32

* The authors did not set the specific number of cores. By default, Gurobi and Cplex generally use all of the cores and threads of the machine.
** Estimated CPU mark for 10 cores.

We first compare the solution generated by our reproduced MIP models with

[1] https://www.passmark.com/

Table 9 Comparison between best known solutions and IP results (new best solutions and new best lower bounds in **bold**, proven optimal solutions in star *).

Instance	Literature results		Warm start WS_0 (MIP models of Bastos et al. (2019))										Warm start WS_7						
			Reported			Our reproduced													
	BKS (Time to end**)	*BLB* (Time to end**)	*Obj*	*LB*	Time to end**	*Obj*	Time to best	Time to end	*LB*	*GAP* (%)	Node of SM	Node of CM	*Obj*	Time to best	Time to end	*LB*	*GAP* (%)	Node of $APRA^{WT}$	Node of APRA
1	651.20 (21,226[1])	651.20 (21,226[1])	651.20	651.20	21,226	651.20*	4,229	19,670	651.20	0.00	91,557	90,609	651.20*	303	1,983	651.20	0.00	1,146	10,811
2	1,128.00 (44,258[1], 2,577[2])	1,115.80 (44,258[1])	1,128.00	1,111.60	44,258	**1,125.60**	30,082	86,400	1,116.20	0.00	9,705	1	**1,125.60***	2,947	25,358	**1,125.60**	0.00	627	22,879
3	761.60 (44,258[1], 2,577[2])	758.60 (44,258[1])	761.60	758.60	44,258	761.60	10,584	86,400	758.60	0.00	644,322	88,845	761.60*	1,315	13,561	**761.60**	0.00	32,952	50,736
4	1,151.60 (44,258[1])	1,143.20 (44,258[1])	1,151.60	1,143.20	44,258	1,151.60	23,864	86,400	1,142.80	0.14	83,928	1	**1,151.00**	21,138	86,400	**1,150.00**	0.09	744,351	326,359
5	624.00 (4,227[1], 62[2])	624.00 (4,227[1])	624.00	624.00	4,227	624.00*	1,073	5,196	624.00	0.00	12,453	12,327	624.00*	286	521	624.00	0.00	679	2,275
6	792.60 (10,082[1], 2,577[2])	792.60 (10,082[1])	792.60	792.60	10,082	792.60*	6,979	11,111	792.60	0.00	377	2,420	792.60*	1,185	2,130	792.60	0.00	8,468	1
7	1,176.40 (6,209[1], 2,577[2])	1,176.40 (6,209[1])	1,176.40	1,175.20	44,258	1,176.40*	451	3,105	1,176.40	0.00	5,193	23,729	1,176.40*	139	696	1,176.40	0.00	865	1,028
8	4,063.00 (44,258[1])	4,024.41 (44,258[1])	4,063.00	4,023.09	44,258	**4,058.60**	47,334	86,400	4,030.20	0.51	76,826	468	**4,058.60**	3,184	86,400	**4,038.00**	0.51	18,531	1,041
9	20,718.60 (3,866[2])	19,632.00 (44,258[1])	20,904.60	19,621.73	44,258	21,109.40	71,202	86,400	**19,872.80**	6.22	14,949	1	**20,677.40**	85,995	86,400	19,862.40	4.05	36,965	1
10	7,804.60 (2,577[2])	7,687.33 (44,258[1])	7,830.40	7,676.04	44,258	7,882.80	85,023	86,400	**7,696.60**	2.42	1,688	1	**7,799.80**	85,991	86,400	7,680.40	1.34	13,705	1
11	11,491.80 (2,577[2])	10,987.72 (44,258[1])	11,932.00	10,914.57	44,258	12,014.80	48,384	86,400	10,937.60	9.35	1	0	11,630.20	48,000	86,400	10,727.05	5.85	1	1
12	22,707.20 (682[2])	21,886.60 (44,258[1])	24,198.40	21,600.30	44,258	24,776.00	11,068	86,400	21,845.00	13.20	1	1	23,234.20	48,000	86,400	21,686.00	6.16	1	1
13	9,109.80 (345[2])	8,842.80 (44,258[1])	9,114.40	8,842.11	44,258	9,148.80	64,047	86,400	8,863.20	3.15	32,044	2	**9,102.20**	67,091	86,400	**8,869.30**	2.63	5,465	1

** Total computation time reported by the corresponding reference, adjusted following the procedure from Da Silva et al. (2012)

[1] - Bastos et al. (2019), [2] - Guido et al. (2018)

the solution presented in Bastos et al. (2019). From the perspective of the solution quality (best objective value), the results of 5 instances (9,10,11,12,13) are worse than those in the literature, 6 instances (1,3,4,5,6,7) are the same as in the literature and 2 instances (2,8) are better than those in the literature. From the perspective of the quality of the lower bounds, the LB of instance 4 are worse than it in the literature, 4 instances (1,3,5,6) are same to literature and 8 instances (2,7,8,9,10,11,12,13) are better than those in the literature. The reasons of above results are due to the used Gurobi version and the performance difference of the computing machines.

Second, we note that our approach generated new best solutions for 6 out of the 13 tested benchmark instances (2,4,8,9,10,13, note that solutions obtained for instances 1,3,5,6 and 7 are the same as the best known solutions reported in the literature; nevertheless, they were proven to be optimal by our $APRA_{AGC_1\&TC}$ model within an hour). Furthermore, our approach improved the best lower bound for 6 out of the 13 instances (2,3,4,7,8,13). It is worth noting that the optimality of the solution was also proven for instance 2. Although we have not proven the optimality of instances 4 and 8, the gaps are very low (¡ 1%). In particular, although our approach fails to improve the best known solutions for instances 11 and 12 within a running time limit of 24 hours, it outperformed the method proposed in Bastos et al. (2019). The failure of our approach to improve upon the best known solutions for instances 11 and 12 can be attributed to the significantly larger number of patients and planning periods in these instances. Specifically, these instances have approximately 2-5 times more patients and 2-6 times more planning periods compared to the others. We also note that our approach is outperformed by the method proposed in Bastos et al. (2019) in terms of the lower bounds for instances 9, 10, 11 and 12. This can be attributed to the fact, as mentiond in Section 4.3.3., that the quality of the linear program (LP) relaxation bound for aggregated model after constraint aggregation is usally poor compared to the LP relaxation of the original model.

Third, we analyze the performance of various models in terms of the number of branch-and-bound nodes visited after the root node. Specifically, we focus on the $APRA^{WT}_{AGC_1}$ and $APRA_{AGC_1\&TC}$ models from our WS_7, and the SM and CM models from WS_0. On the one hand, we can observe that for the instances that are solved to optimality by WS_7, the number of nodes in the B&C procedure for WS_7 is generally less than WS_0. Specifically, for the instances 1,5,6,7 which are also solved to optimality by WS_0, WS_0 generated on average 27395 nodes in SM and 31727 nodes in CM, while WS_7 generated on average 2790 nodes in APRA^{WT} and 3529 nodes in APRA. For instances 2 and 3 which can not be solved to optimality by the WS_0, WS_0 generated on average 327014 nodes in SM and 44423 nodes in CM, while WS_7 generated on average 16790 nodes in APRA^{WT} and 36808 nodes in APRA. On the other hand, for the instances 4,8,9,10,11,12,13 which can not be solved to optimality by both WS_0 and WS_7, the number of nodes in the B&C procedure for WS_7 is generally more than WS_0. In particular, WS_0 generated on average 71348 nodes in SM and 68 nodes in CM, while WS_7 generated on average 117003 nodes in APRA$^{\text{WT}}$ and 46772 nodes in APRA. These results are explained by the fact that the B&C procedure for $APRA_{AGC_1}^{WT}$ and $APRA_{AGC_1\&TC}$ leads to considerably smaller branch-and-bound trees.

The above comparisons demonstrate that our proposed two-stage optimization approach, featuring the APRA$^{\text{WT}}$ and APRA models, is able to significantly reduce the computation time compared to the MIP model proposed by Bastos et al. (2019). The APRA$^{\text{WT}}$ and APRA models, refined from the IP model proposed by Ceschia & Schaerf (2011) by our proposed 6 rules and constraint aggregation, have fewer variables and constraints than those of Bastos et al. (2019). In particular, $APRA_{AGC_1}^{WT}$ and $APRA_{AGC_1\&TC}$, the best version of our proposed models, achieve reductions by 97.29% and 79.47% for variables, and 99.89% and 92.44% for constraints, respectively. This substantial simplification aligns with the general principle that smaller models are typically easier to solve than their larger counterparts. Consequently, the number of nodes in the B&C procedure is less than that of Bastos et al. (2019), which leads to a significant reduction in the computation time.

Finally, we present a breakdown of the cost of our best solutions into different objective components in Table 10. It reports for each instance, the total penalty (Cost), the penalty associated with gender policy violations (Gen.), the penalty associated with age policy violations (Age), the penalty not attending to the needed treatment properties (Ned. prop.), the penalty related to single policy violations (Sng.), the penalty for failing to assign a patient to a room with his/her preferred capacity (Room pref.), the penalty incurred by not assigning a patient to the appropriate department (Dept.), the penalty incurred by not accounting for the prioritized specialism (Spec.), the penalty not attending to the preferred treatment properties (Pref. prop.), and the penalty related to transfers policy violations (Trs.).

Most penalties in instances 1-6 are caused by not being able to satisfy room capacity preferences, and specialisms and room properties preferences also contribute in the same cases, as reported by Range et al. (2014); Bastos et al. (2019). In addition to the above penalties, department violations appeared for instance 7. For instances 8-13, preferred treatment properties violations account for most of the cost, and department, Specialism, and preferred room capacity violations have been consistently detected. Moreover, age policy violations appeared for instance 9 and 13, and gender violations appear for instances 9, 10, 11, 12 and 13. Finally, we note that the transfer violations were reported in instances 8, 9 and 13.

Table 10
Breakdown of the cost components for the best solutions.

Instance	Cost	Gen.	Age	Ned. pref.	Sng.	Room pref.	Dept.	Spec.	Pref. prop.	Trs.
1	651.2	0.0	0.0	0.0	0.0	651.2	0.0	0.0	0.0	0.0
2	1125.6	0.0	0.0	0.0	0.0	1113.6	0.0	12.0	0.0	0.0
3	761.6	0.0	0.0	0.0	0.0	753.6	0.0	8.0	0.0	0.0
4	1151.0	0.0	0.0	0.0	0.0	1040.0	0.0	75.0	36.0	0.0
5	624.0	0.0	0.0	0.0	0.0	624.0	0.0	0.0	0.0	0.0
6	792.6	0.0	0.0	0.0	0.0	789.6	0.0	3.0	0.0	0.0
7	1176.4	0.0	0.0	0.0	0.0	730.4	20.0	158.0	268.0	0.0
8	4058.6	0.0	0.0	0.0	0.0	1433.6	212.0	869.0	1522.0	22.0
9	20677.4	340.0	4500.0	1010.0	0.0	2702.4	282.0	1124.0	10554.0	165.0
10	7799.8	15.0	0.0	0.0	0.0	2964.8	2.0	486.0	4332.0	0.0
11	11630.2	10.0	0.0	5.0	0.0	4327.2	9.0	959.0	6320.0	0.0
12	23234.2	585.0	0.0	195.0	0.0	4823.2	280.0	1751.0	15600.0	0.0
13	9102.2	25.0	30.0	35.0	0.0	2091.2	655.0	1730.0	4470.0	66.0

5.3 Application to the original PAS problem

As mentioned in Section 2.1, various static PAS problems have been studied in the literature. The differences are the treatment of SC1-SC4 and SC9 constraints. Our proposed method can solve these static variants by decreasing the number of the soft constraint and adjusting the domain of the patient-room assignment variables according to the specific problem definition. Different with the standard PAS problem we solved, in the original PAS problem proposed by Demeester et al. (2010), the former four constraints are hard constraints, which are not allowed to be violated. In order to solve the original PAS problem, we use our proposed two-stage optimization approach and modify the APRA$^{\mathrm{WT}}$ and APRA models by limiting the set of rooms that can be assigned to each patient. Specifically, we use $\mathcal{R}_p \in \mathcal{R}$, which is defined as the set of rooms that can be assigned to patient p without violating the constraints SC1-SC4. The modified $APRA^{WT}$ and $APRA$ models are formulated as follows:

$$\text{Modified } \boldsymbol{APRA^{WT}}\text{:} \quad \text{Min} \quad S = \sum_{p\in\mathcal{P}^E} \sum_{r\in\mathcal{R}_p} C'_{pr} x_{pr} \tag{52}$$

s.t. Constraints (30), (34), where $\mathcal{R}_p$ is used instead of $\mathcal{R}$

Constraints (23)

$$\sum_{p\in\mathcal{P}|d\in\mathcal{D}_p, r\in\mathcal{R}_p} x_{pr} \le Q_r, \quad \forall d\in\mathcal{D}, r\in\mathcal{R} \tag{53}$$

$$\lambda^F_{rd} f_{rd} \ge \sum_{p\in\mathcal{F}|d\in\mathcal{D}_p, r\in\mathcal{R}_p} x_{pr} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^M_D \tag{54}$$

$$\lambda^M_{rd}(1-f_{rd}) \ge \sum_{p\in\mathcal{M}|d\in\mathcal{D}_p, r\in\mathcal{R}_p} x_{pr} \quad \forall d\in\mathcal{D}, r\in\mathcal{R}^M_D \tag{55}$$

$$
\text{Modified } \boldsymbol{APRA}: \quad \text{Min} \quad S = \sum_{p \in \mathcal{P}^E, r \in \mathcal{R}_p, d \in \mathcal{D}_p} C_{prd} \cdot x_{prd} + \sum_{p \in \mathcal{P}^E | LOS_p \geq 2, r \in \mathcal{R}_p, d \in \mathcal{D}_p \setminus \{DD_p - 1\}} W_{Tr} \cdot t_{prd} \tag{56}
$$

s.t. Constraints (15), (20), (21), (25), where $\mathcal{R}_p$ is used instead of $\mathcal{R}$

Constraints (53), (54), (55), where x_{prd} is used instead of x_{pr}

Constraints (23)

The computational results are summarized in Table 11. Since SC1-SC4 are considered to be hard constraints, the corresponding penalties (Gen., Age, Sng. and Ned. Pref.) are equal to zero and are not reported. Note that most studies treat SC1-SC4 as soft constraints, and therefore the corresponding problems are relaxations of the original PAS problem. Consequently, the best lower bounds in those studies can be used as the best known lower bounds BLB of the original PAS problem. The best known solutions BKS are similarly derived from the results in the literature without incurring penalties of SC1-SC4. The symbol "-" is used to indicate that the instance is infeasible for the original PAS problem, or the result is not available in the literature.

Table 11
Results on the benchmark instances for the original PAS problem (new best solutions and new best lower bounds in **bold**, proven optimal solutions in star *).

Instance	Literature results		Two-stage optimization approach											
	BKS (Time to end**)	*BLB* (Time to end**)	*Obj*	Time to best	Time to end	*LB*	*GAP* (%)	Node of APRA^{WT}	Node of APRA	Breakdown of the *Obj* components: Room pref.	Dept.	Spec.	Pref. prop.	Trs.
1	651.20 (21,226[1])	651.20 (21,226[1])	651.20*	272	5,370	651.20	0.00	5,872	6,732	651.2	0.0	0.0	0.0	0.0
2	1,128.00 (44,258[1], 2,577[2])	1,115.80 (44,258[1])	**1,125.60***	7,638	24,372	**1,125.60**	0.00	10,447	8,356	1,113.6	0.0	12.0	0.0	0.0
3	761.60 (44,258[1], 2,577[2])	758.60 (44,258[1])	761.60*	2,021	12,864	**761.60**	0.00	46,752	62,533	753.6	0.0	8.0	0.0	0.0
4	1,151.60 (44,258[1])	1,143.20 (44,258[1])	**1,151.00**	35,818	86,400	**1,150.00**	0.09	144,679	553,384	1040.0	0.0	75.0	36.0	0.0
5	624.00 (4,227[1], 62[2])	624.00 (4,227[1])	624.00*	199	752	624.00	0.00	670	5,603	624.0	0.0	0.0	0.0	0.0
6	792.60 (10,082[1], 2,577[2])	792.60 (10,082[1])	792.60*	462	1,019	792.60	0.00	45	1	789.6	0.0	3.0	0.0	0.0
7	1,176.40 (6,209[1], 2,577[2])	1,176.40 (4,209[1])	1,176.40*	36	486	1,176.40	0.00	321	5,206	730.4	20.0	158.0	268.0	0.0
8	4,063.00 (44,258[1])	4,024.41 (44,258[1])	**4,058.60**	9,655	86,400	**4,039.60**	0.47	9,137	3,020	1,433.6	214.0	871.0	1,518.0	22.0
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	7,804.60 (2,577[2])	7,687.33 (44,258[1])	**7,793.80**	43,200	86,400	**7,719.60**	0.96	1,298	1	2,948.8	4.0	481.0	4,360.0	0.0
11	11,536.20 (654[2])	10,987.72 (44,258[1])	11,836.60	43,200	86,400	10727.05	7.73	268	0	4,361.6	16.0	963.0	6,496.0	0.0
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	8,842.80 (44,258[1])	**9,093.60**	67,866	86,400	**8,912.40**	2.03	66,457	441	2,061.6	627.0	1,695.0	4,556.0	154.0

** Total computation time reported by the corresponding reference, adjusted following the procedure from Da Silva et al. (2012)
[1] - Bastos et al. (2019), [2] - Guido et al. (2018)

From Table 11, we observe that our approach computed 5 out of 13 new best solutions on the tested benchmark instances (2, 4, 8, 10, 13, solutions obtained for instances 10, 13 are better than our new found solutions in Table 9). Our approach proved the optimality of 6 out of 13 solutions (1, 2, 3, 5, 6, 7). Moreover, our approach improved the best lower bound for 6 out of the instances (2, 3, 4, 8, 10, 13, lower bounds obtained for instances 8, 10, 13 are better than our new found lower bounds in Table 9). Note that instances 9 and 12 are infeasible in the original PAS problem. The reason is that the number of elective patients exceeds the capacity of the rooms allowed for them.

6 Conclusion

The patient admission scheduling (PAS) problem is a significant planning task in hospital management. Due to the high-quality performance of exact algorithms in the literature, in this paper, we focused on ways to improve the efficiency of solving the IP model of the PAS problem using better model formulations. We employed a two-stage exact method that decomposes the PAS problem into two separate problems, including the patient-room assignment (PRA) subproblem and the patient-bed assignment (PBA) subproblem. To solve the PRA subproblem, we applied a warm start approach in which we solve the APRA^{WT} model to generate a high-quality feasible solution and then use the obtained solution as a warm start to the APRA model.

We proposed two aggregated gender policy constraints AGC_0, AGC_1, and aggregated patient transfer constraint ATC, and generated 4 APRA^{WT} models and 8 APRA models. We analyzed the advantages and disadvantages of these models and found the most appropriate model, using the aggregated gender constraint AGC_0 for APRA^{WT} and APRA, not the ATC constraint for APRA.

Our approach generated new best solutions for 6 out of the 13 benchmark instances from a publicly available repository, and proved the optimality of the solution for one of these 6 instances. Moreover, for 5 other instances, we obtained the known optimal solutions in a short time compared to the methods in the literature. Finally, we also applied our approach to the original PAS problem and performed computational experiments on the same 13 benchmark instances. We obtained 5 new best solutions, 6 new best lower bounds, and proved optimality for 6 instances.

To further exploit the new reduced models, we suggest the following directions for future research: (1) Design dedicated branch-and-bound algorithms to improve solving performance while guaranteeing optimality. (2) Design matheuristic algorithms that exploit mathematical programming techniques in a metaheuristic framework. (3) Investigate the dynamic PAS problem by appropriately adjusting the proposed models to solve real-world situations.

Acknowledgment

We are grateful to the reviewers for their insightful comments, which helped us improve the paper significantly. This work was partially supported by the National Natural Science Foundation of China (No. 71971172, No. 72371200), the Fundamental Research Funds for the Central Universities (No. D5000210834), the Featured Humanities Development Plan of Northwestern Polytechnical University – Youth Innovation Capability Training Program (No. 23GH030611), and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (No. CX2022057). Support from China Scholarship Council (CSC, Grant No. 202206290116) for the first author is acknowledged.

References

Abdalkareem, Z. A., Al-Betar, M. A., Amir, A., Ehkan, P., Hammouri, A. I., & Salman, O. H. (2022). Discrete flower pollination algorithm for patient admission scheduling problem. *Computers in Biology and Medicine*, 141, 105007.

Abu Doush, I., Al-Betar, M. A., Awadallah, M. A., Hammouri, A. I., Al-Khatib, R. M., ElMustafa, S., & ALkhraisat, H. (2018). Harmony search algorithm for patient admission scheduling problem. *Journal of Intelligent Systems*, 29(1), 540–553.

Alidaee, B. (2014). Zero duality gap in surrogate constraint optimization: A concise review of models. *European Journal of Operational Research*, 232(2), 241–248.

Bastos, L. S., Marchesi, J. F., Hamacher, S., & Fleck, J. L. (2019). A mixed integer programming approach to the patient admission scheduling problem. *European Journal of Operational Research*, 273(3), 831–840.

Benchimol, P., Desaulniers, G., & Desrosiers, J. (2012). Stabilized dynamic constraint aggregation for solving set partitioning problems. *European Journal of Operational Research*, 223(2), 360–371.

Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W., & Berghe, G. V. (2012). One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics*, 18(3), 401–434.

Bolaji, A. L., Bamigbola, A. F., Adewole, L. B., Shola, P. B., Afolorunso, A., Obayomi, A. A., Aremu, D. R., & Almazroi, A. A. A. (2022). A room-oriented artificial bee colony algorithm for optimizing the patient admission scheduling problem. *Computers in Biology and Medicine*, 148, 105850.

Bolaji, A. L., Bamigbola, A. F., & Shola, P. B. (2018). Late acceptance hill climbing algorithm for solving patient admission scheduling problem. *Knowledge-Based Systems*, 145, 197–206.

Ceschia, S. & Schaerf, A. (2011). Local search and lower bounds for the patient admission scheduling problem. *Computers & Operations Research*, 38(10), 1452–1463.

Ceschia, S. & Schaerf, A. (2012). Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Artificial intelligence in medicine*, 56(3), 199–205.

Ceschia, S. & Schaerf, A. (2016). Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. *Journal of Scheduling*, 19(4), 377–389.

Da Silva, G. C., Bahiense, L., Satoru Ochi, L., & Boaventura-Netto, P. O.

(2012). The dynamic space allocation problem: Applying hybrid grasp and tabu search metaheuristics. *Computers & Operations Research*, 39(3), 671–677.

Demeester, P., De Causmaecker, P., & Vanden Berghe, G. (2008). Applying a local search algorithm to automatically assign patients to beds. *Proceedings of the 22nd conference on quantitative methods for decision making (Orbel 22)*, 35–36.

Demeester, P., Souffriau, W., De Causmaecker, P., & Berghe, G. V. (2010). A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine*, 48(1), 61–70.

Elhallaoui, I., Desaulniers, G., Metrane, A., & Soumis, F. (2008). Bi-dynamic constraint aggregation and subproblem reduction. *Computers & Operations Research*, 35(5), 1713–1724.

Elhallaoui, I., Villeneuve, D., Soumis, F., & Desaulniers, G. (2005). Dynamic aggregation of set-partitioning constraints in column generation. *Operations Research*, 53(4), 632–645.

Ermoliev, Y. M., Kryazhimskii, A. V., & Ruszczyński, A. (1997). Constraint aggregation principle in convex optimization. *Mathematical Programming*, 76(3), 353–372.

Evans, J. R. (1983). A network decomposition/aggregation procedure for a class of multicommodity transportation problems. *Networks. An International Journal*, 13(2), 197–205.

Glover, F. (2003). Tutorial on surrogate constraint approaches for optimization in graphs. *Journal of Heuristics*, 9, 175–227.

Guido, R. (2023). Patient admission scheduling problems with uncertain length of stay: optimization models and an efficient matheuristic approach. *International Transactions in Operational Research*.

Guido, R., Groccia, M. C., & Conforti, D. (2018). An efficient matheuristic for offline patient-to-bed assignment problems. *European Journal of Operational Research*, 268(2), 486–503.

Hammouri, A. & Alweshah, M. (2017). Biogeography based optimization with guided bed selection mechanism for patient admission scheduling problems. *International Journal of Soft Computing*, 12, 103–111.

Hammouri, A. I. (2022). A modified biogeography-based optimization algorithm with guided bed selection mechanism for patient admission scheduling problems. *Journal of King Saud University - Computer and Information Sciences*, 34(3), 871–879.

Hammouri, A. I. & Alrifal, B. (2014). Investigating biogeography-based optimisation for patient admission scheduling problems. *Journal of Theoretical & Applied Information Technology*, 70(3), 413–421.

Khurana, A. & Murty, K. G. (2012). How effective is aggregation for solving 0–1 models? *OPSEARCH*, 49(1), 78–85.

Kifah, S. & Abdullah, S. (2015). An adaptive non-linear great deluge algorithm for the patient-admission problem. *Information Sciences*, 295, 573–585.

Litvak, E. & Bisognano, M. (2011). More patients, less payment: increasing hospital efficiency in the aftermath of health reform. *Health Affairs*, 30(1), 76–80.

Lusby, R. M., Schwierz, M., Range, T. M., & Larsen, J. (2016). An adaptive large neighborhood search procedure applied to the dynamic patient admission scheduling problem. *Artificial intelligence in medicine*, 74, 21–31.

Poirion, P.-L. (2019). Optimal constraints aggregation method for ilp. *Discrete Applied Mathematics*, 262, 148–157.

Porumbel, D. & Clautiaux, F. (2017). Constraint aggregation in column generation models for resource-constrained covering problems. *INFORMS Journal on Computing*, 29(1), 170–184.

Ram, B., Karwan, M. H., & Babu, A. (1988). Aggregation of constraints in integer programming. *European Journal of Operational Research*, 35(2), 216–227.

Range, T. M., Lusby, R. M., & Larsen, J. (2014). A column generation approach for solving the patient admission scheduling problem. *European Journal of Operational Research*, 235(1), 252–264.

Rogers, D. F., Plante, R. D., Wong, R. T., & Evans, J. R. (1991). Aggregation and disaggregation techniques and methodology in optimization. *Operations Research*, 39(4), 553–582.

Saddoune, M., Desaulniers, G., Elhallaoui, I., & Soumis, F. (2011). Integrated airline crew scheduling: A bi-dynamic constraint aggregation method using neighborhoods. *European Journal of Operational Research*, 212(3), 445–454.

Trapp, A. C. & Prokopyev, O. A. (2015). A note on constraint aggregation and value functions for two-stage stochastic integer programs. *Discrete Optimization*, 15, 37–45.

Turhan, A. M. & Bilgen, B. (2017). Mixed integer programming based heuristics for the patient admission scheduling problem. *Computers & Operations Research*, 80, 38–49.

Vancroonenburg, W., De Causmaecker, P., & Vanden Berghe, G. (2016). A study of decision support models for online patient-to-room assignment planning. *Annals of Operations Research*, 239(1), 253–271.

Vancroonenburg, W., Goossens, D., & Spieksma, F. (2011). On the complexity of the patient assignment problem. *Tech. rep., KAHO Sint-Lieven, Gebroeders De Smetstraat 1, Gent, Belgium.*

Zhang, J., Dridi, M., & Moudni, A. E. (2021). An approximate dynamic programming approach to the admission control of elective patients. *Computers & Operations Research*, 132(1–2), 105259.

Zhang, K.-S., Han, Z.-H., Gao, Z.-J., & Wang, Y. (2019). Constraint aggregation for large number of constraints in wing surrogate-based optimization. *Structural and Multidisciplinary Optimization*, 59(2), 421–438.

Zhu, Y.-H., Toffolo, T. A. M., Vancroonenburg, W., & Vanden Berghe, G. (2019). Compatibility of short and long term objectives for dynamic patient admission scheduling. *Computers & Operations Research*, 104, 98–112.