ONLINE CAPACITY PLANNING FOR REHABILITATION TREATMENTS: AN APPROXIMATE DYNAMIC PROGRAMMING APPROACH

Ingeborg A. Bikker [a,b,c,1], Martijn R.K. Mes [d], Antoine Sauré [e], Richard J. Boucherie [a,c]

a Center for Healthcare Operations Improvement and Research (CHOIR), University of Twente, Drienerlolaan 5, 7500 AE Enschede, the Netherlands.
b Department of Healthcare Logistics, Sint Maartenskliniek, Hengstdal 3, 6574 NA Ubbergen (Nijmegen), the Netherlands.
c Stochastic Operations Research, Department of Applied Mathematics, University of Twente, Drienerlolaan 5, 7500 AE Enschede, the Netherlands.
d Department Industrial Engineering and Business Information Systems (IEBIS), University of Twente, Drienerlolaan 5, 7500 AE Enschede, the Netherlands.
e Telfer School of Management, University of Ottawa, 55 Laurier Avenue East, Ottawa, ON, Canada K1N 6N5

Abstract

We study an online capacity planning problem in which arriving patients require a series of appointments at several departments, within a certain access time target.

This research is motivated by a study of rehabilitation planning practices at the Sint Maartenskliniek hospital (the Netherlands). In practice, the prescribed treatments and activities are typically booked starting in the first available week, leaving no space for urgent patients who require a series of appointments at a short notice. This leads to rescheduling of appointments or long access times for urgent patients, which has a negative effect on the quality of care and on patient satisfaction.

We propose an approach for allocating capacity to patients at the moment of their arrival, in such a way that the total number of requests booked within their corresponding access time targets is maximized. The model considers online decision making regarding multi-priority, multi-appointment, and multi-resource capacity allocation. We formulate this problem as a Markov decision process (MDP) that takes into account the current patient schedule, and future arrivals. We develop an approximate dynamic programming (ADP) algorithm to obtain approximate optimal capacity allocation policies. We provide insights into the characteristics of the optimal policies and evaluate the performance of the resulting policies using simulation.

Keywords: operations research, healthcare logistics, rehabilitation treatment planning, online capacity planning, Markov decision process, approximate dynamic programming, simulation.

---

[1] Corresponding author. E-mail: i.a.bikker@utwente.nl.

1. Introduction

Worldwide, millions of patients are admitted to rehabilitation facilities each year as a consequence of accidents, sickness or congenital disorders. In rehabilitation, a patient is assisted in improving or recovering lost functions after an accident, illness or injury. A patient is treated by a multidisciplinary team of practitioners, including, e.g., physiotherapists and occupational therapists, under the lead of a rehabilitation doctor. In the Netherlands, over 90,000 patients are treated in rehabilitation facilities each year [1]. A rehabilitation treatment consists of an intake appointment with a doctor and a series of treatment sessions over several weeks or months, either on an inpatient or an outpatient basis.

The best quality of care is delivered when the right care is provided at the right time [5]. Rehabilitation care professionals claim that, amongst others, a short access time [26], a simultaneous start of sessions from the different involved disciplines and the continuity of the rehabilitation process [3] should be guaranteed. Access time targets can vary widely depending on the type of patient: some patients have to start treatment within 24 hours whereas others can wait for several weeks.

This paper presents a methodology for capacity planning of rehabilitation treatments in an online fashion, i.e., capacity is booked for multiple weeks in advance as the treatment requests arrive. The method is designed in such a way that compliance to access time targets (concerning the recommended maximum number of weeks between consultation and start of the treatment) is optimized across all patients. The performance of the proposed methodology is evaluated for a case study based on data of the Sint Maartenskliniek, a major rehabilitation center in the Netherlands.

Planning an appointment series for a patient is the act of determining a future starting week of his or her treatment, setting the weekdays and time slots for each appointment, and assigning a therapist to each appointment. This appointment planning process is complex, because many constraints and preferences are involved, e.g., the availability of several therapists (often available part time) and the patient's preference to combine appointments for different disciplines on the same day. Another factor that complicates appointment planning is the fact that appointments are often booked as the treatment requests arrive (so-called *online* appointment planning). This is done as a service to the patient, who then immediately knows when his or her treatment will start. However, this implies that capacity planning decisions have to be made before we know which other requests need to be scheduled. In practice, treatments are often booked to start in the first week with enough capacity available. The risk is that there might be no space left to treat urgent or resource intensive patients within their corresponding access time targets.

The planning problem we consider in this study can be described as follows. An incoming request for a patient consists of weekly capacity requirements for different disciplines over a consecutive number of weeks and an access time target, which states the number of weeks within which the treatment should start. Each time a request comes in, the required capacity has to

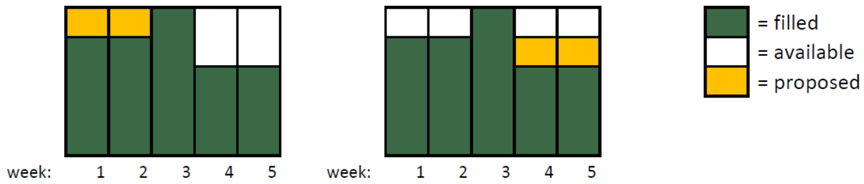

Figure 1: Example of a planning decision for one patient, who can start either in week 1 (left) or week 4 (right).

be booked immediately for all the weeks and disciplines associated with the treatment. Since the treatment has to be delivered in consecutive weeks and has to start in the same week for all involved disciplines, our decision entails determining a starting week for each patient on the moment of his/her arrival. Note that the decision on the exact days, time slots and therapists is not taken into account in this study. The treatment can only take place if there is enough capacity available in each week for each discipline. If there is not enough capacity in the booking horizon, a patient may be diverted to another hospital. No changes can be made in already allocated capacity, i.e., no rescheduling is allowed. The objective of this problem is to maximize the number of requests booked within their access time targets.

We illustrate the problem with the following example. Suppose we have an existing schedule, where capacity is partly allocated. For simplicity, we consider the availability of one discipline with a single therapist on a weekly basis (see Figure 1 for a schematic representation). A request for one time slot of treatment during two consecutive weeks arrives, with an access time target of four weeks. According to the current schedule, there is at least one time slot available in each week, except for the third week. Thus, we could choose to start the treatment in week 1, but we could also postpone the start until week 4 such that future urgent requests could be booked within their access time target of one week. Determining which starting week is optimal depends on the nature of the arrival process of future requests.

In this study, we develop an online capacity planning algorithm that provides guidelines for allocating capacity to incoming requests, taking into account the arrival process of future requests, and taking into account that later arriving requests may have shorter access time targets. This is the first model that considers online decision making regarding multi-priority, multi-appointment, and multi-resource capacity allocation. Our contribution is threefold. First, we define a Markov decision process (MDP) to model our planning problem. Due to the curse of dimensionality, the MDP cannot be solved analytically for realistically sized problems. Therefore, our second contribution is the development of an approximate dynamic programming (ADP) approach, using affine value function approximation and column generation, to obtain approximate solutions for the MDP. To find a suitable approximation of the value function, we provide a founded choice of basis functions for the considered problem. Third, we provide insights into the characteristics of the optimal policy. The model is applied to a case study based on data of the Spinal Cord Injury Unit of the Sint Maartenskliniek in the Netherlands. Simulation results show that the use of this method in practice could substantially increase the number of patients for whom treatment starts within their access time targets.

The remainder of this paper is organized as follows. Section 2 provides an overview of the related literature. In Section 3, we explain our MDP model. We develop an ADP approach to solve the MDP model in Section 4. Section 5 presents the results, followed by conclusions and discussion in Section 6.

2. Related literature

To place our contribution in the context of previous work, we discuss the related literature on appointment planning in healthcare. We explain that our type of problem can be modeled as an MDP and we discuss the application of ADP for obtaining approximate solutions to it.

In this paper, we consider an *online* planning problem, in which patients get a direct response to their treatment request in the form of a starting week. *Offline* planning involves collecting treatment requests until all demand for a certain period is known and then book all requests. *Semi-online planning* entails collecting requests and booking requests at the end of each day.

Appointment planning in healthcare has received considerable attention in the literature, especially offline scheduling. Two comprehensive surveys are provided in [4] and [15]. The literature has mostly focused on planning single appointments on a particular day for an individual service provider [4]. Some research has been done on planning series of multi-disciplinary appointments [17]. Most notably, Schimmelpfeng et al. [27] developed an offline decision support system for multi-disciplinary planning in rehabilitation facilities. The authors formulated a mixed integer linear programming model that is decomposed into a hierarchical three-stage model to reduce computational difficulties. For online planning, comprehensive reviews can be found in [23, 31]. An approach for online planning of rehabilitation appointment series is proposed by Braaksma et al. [3]. In this paper, the access time targets of all patients are similar, so no capacity has to be reserved for future urgent patients.

Online and semi-online planning can significantly benefit from anticipating future treatment requests when booking the current treatment request [20]. Taking future requests into account in online planning involves optimization over time, which can be represented by a sequential decision problem, for example by an MDP. Due to the curses of dimensionality, these models can often not be solved exactly. One technique that can be used to derive approximate solutions for the MDP is ADP. A comprehensive explanation and overview of the various ADP techniques is given in [22]. Some practical examples are described in [19]. ADP is used in, e.g., resource capacity planning [9, 29, 30], inventory control [32], and transportation [33]. The application of ADP in healthcare is relatively new. It has been used in ambulance planning (e.g., [18, 28]), admission control (e.g., [16]) and patient planning [9, 12, 21, 25]. In the work of [9, 12, 21], planning one appointment in a semi-online manner is done using ADP. In [12], the appointment should take place in a certain interval. In [25], series of radiation appointments on consecutive days are considered, where all appointments require the same resource. Booking decisions are taken at the end of each day. We have not found any papers on multi-disciplinary appointment planning in an online fashion, using ADP.

An ADP approach for solving MDPs usually takes one of the common MDP solution methods as a basis: value iteration, policy iteration or linear programming. The value iteration method is mostly used as a basis for the ADP approach. However, since De Farias and Van Roy [6] introduced ADP concepts to the linear programming method in 2003, this method has received attention as a basis for ADP. Two major problems may occur with the common linear programming method as a consequence of the curses of dimensionality. First, the number of decision variables (equal to the number of states) may be too large, which can be solved introducing basis functions to approximate the representation of the value function in the MDP formulation; examples of this can be found in [10, 34]. Second, the number of constraints (equal to the number of states times the number of actions) may be too large. The latter problem can be solved using Monte Carlo samples of the constraints [7], by allowing constraints to be represented compactly [14], or by using problem specific approaches making use of constraint generation/column generation methods [11, 13, 21, 25]. In the planning problem of this paper, we face both curses of dimensionality, and deal with them using value function approximation and column generation respectively.

Our contribution is the development of a method for online capacity planning over multiple weeks and involving multiple disciplines and priorities, taking into account future requests. In existing approaches, all appointments require the same resource, whereas we study the interrelatedness of multiple disciplines with their own capacity limits and their combined influence on the optimal policy. Also, healthcare logistics literature merely contains approaches that save up requests and book them at the end of one day, which entails more prior knowledge than the planning problem we consider where patients get a direct response to their treatment request. Our approach is novel in the area of rehabilitation treatment, as existing approaches in rehabilitation only use offline planning, or online planning but without taking future requests into account. We use a linear programming approach as a basis to solve the MDP, and develop an ADP approach for this problem, using affine value function approximation and column generation, to derive approximate solutions. In order to find a suitable approximation of the value function, we provide a founded choice of basis functions for the considered problem.

3. Markov decision model

The planning problem we consider deals with booking capacity for each incoming treatment request at the moment of arrival. Since the treatment has to take place over a number of consecutive weeks and has to start in the same week for all involved disciplines, our decision entails determining a starting week for each patient at the moment of his/her arrival. The objective is to maximize the number of requests booked within their access time targets.

We model this problem as a Markov decision process (MDP). MDPs have proven to be useful as models for sequential decision problems with stochastic characteristics that have the Markov property (i.e., future states and decisions are independent from past states and decisions, given the present state of the system). At each decision epoch in an MDP, the system's state is observed and one decision (action) is chosen. Based on the current state and action, the probabilities of reaching any possible system state in the next decision epoch are known, as well as the expected cost to be incurred given the current state and action. This decision process is repeated over and over. MDPs are comprehensively discussed by Puterman [24] and more recently in Boucherie and Van Dijk [2].

We consider an MDP with an infinite horizon, since the considered system has no predetermined time of extinction and we are interested in stationary planning policies. Let $\mathcal{D}$, $\mathcal{W}$, $\mathcal{K}$ and $\mathcal{R}$ be sets containing the disciplines, weeks in the booking horizon, time slots in a week, and types of requests respectively. The cardinalities of the sets are denoted by the corresponding capitals, e.g., $\mathcal{W} = \{1, \ldots, W\}$. Typical elements are denoted by the corresponding lower case characters. The decision epochs, states, actions, transition probabilities and costs are described in the following subsections and summarized in Table 1.

3.1. Decision epochs and booking horizon

We use two different time scales: one related to the weeks of the booking horizon, and the other related to the decision epochs. For the booking horizon, we consider W weeks. With respect to the decision epochs, we divide the current week into K time intervals, at the end of which a decision is taken. K is chosen in such a way that the probability of two or more requests arriving in the same interval is negligible; so K depends on the patient arrival distribution (see Appendix A). It is possible that no treatment request arrives in an interval. In that case, we immediately proceed to the next decision epoch.

3.2. States

We denote the set of all possible states of the system by $\mathcal{S}$. A state is represented by a triple $s = \langle y(s), r(s), k(s)\rangle$, where y is a $\mathcal{D} \times \mathcal{W}$-matrix of which each cell $y_{dw}(s)$ contains the number of time slots that are already booked for discipline d in week w in state s. The total number of time slots that can be booked per week for a discipline d is denoted by q_d. Parameters $r(s) \in \mathcal{R}$ and $k(s) \in \mathcal{K}$ denote the current request and the current decision epoch in state s respectively. A request is either empty (meaning that no request arrived in the decision epoch) or consists of a triple $r(s) = \langle f^r, \ell^r, b^r\rangle$, where b^r is the access time target in weeks, and f^r and ℓ^r are vectors in $\mathbb{N}^{\mathcal{D}}$ such that for each $d \in \mathcal{D}$:

- f^r_d is the number of consecutive weeks in which treatment from discipline d is required;
- ℓ^r_d is the number of required time slots from discipline d in each week of the treatment.

3.3. Actions

An action a represents the week in the booking horizon in which the current requested treatment starts. In case of an empty request or if a request is diverted to another hospital, we set $a = 0$. Let $A_s \subseteq \{0, 1, \ldots, W\}$ denote the set of feasible actions in state $s \in \mathcal{S}$ and let $q \in \mathbb{N}^{\mathcal{D}}$ contain the weekly capacities of the disciplines. The action $a = 0$ is always feasible. Given state $s = \langle y(s), r(s), k(s) \rangle$, an action $a > 0$ is feasible if the request $r(s) = r$ is non-empty and

- the final treatment week cannot be scheduled beyond the last week of the booking horizon: $a + f_d^r - 1 \leq W \ \forall d \in \mathcal{D}$;
- there must be sufficient capacity for each discipline in all weeks of the treatment plan $\ell_d^r \leq q_d - y_{dw}(s) \ \forall d \in \mathcal{D}, w \in \{a, \ldots, a + f_d^r - 1\}$.

Note that we assume that a patient's treatment cannot start in the same week as his/her arrival. However, this assumption can easily be relaxed.

3.4. Transition probabilities

When an action a is chosen (i.e., the current request is booked or diverted), the only stochastic element in the transition to the next state is the new request $r(s') = r$ with probability $\mathbb{P}(r)$. For states with $k < K$, the state transitions $\langle y(s), r(s), k(s) \rangle \longrightarrow \langle y(s'), r(s'), k(s) + 1 \rangle$ are further described by the following equations. For actions $a > 0$ holds:

$$y_{dw}(s') = \begin{cases} y_{dw}(s) + \ell_d^r & \forall d, w \in \{a, \ldots, a + f_d^r - 1\}, \\ y_{dw}(s) & \forall d, w \notin \{a, \ldots, a + f_d^r - 1\}. \end{cases} \tag{1}$$

For action $a = 0$ holds, in case $k < K$:

$$y_{dw}(s') = y_{dw}(s) \quad \forall d, w. \tag{2}$$

If $k = K$, the week indices shift one week in the next state, so week w in the current state becomes week $w - 1$ at the subsequent state. At the end of the booking horizon, a new week is added in which no treatments have yet been booked. The state transitions $\langle y(s), r(s), K \rangle \longrightarrow \langle y(s'), r(s'), 1 \rangle$ are further described by the following equations. For actions $a > 0$ holds:

$$y_{dw}(s') = \begin{cases} 0 & \forall d, w = W, \\ y_{d(w+1)}(s) + \ell_d^r & \forall d, w \in \{\max(a-1, 1), \ldots, a + f_d^r - 2\}, w < W, \\ y_{d(w+1)}(s) & \forall d, w \notin \{\max(a-1, 1), \ldots, a + f_d^r - 2\}, w < W. \end{cases} \tag{3}$$

For action $a = 0$ holds, in case $k = K$:

$$y_{dw}(s') = \begin{cases} 0 & \forall d, w = W, \\ y_{d(w+1)}(s) & \forall d, w < W. \end{cases} \tag{4}$$

3.5. Immediate costs

If action a is chosen in state s, a cost $c(s,a)$ is incurred, consisting of a penalty for exceeding the access time target, a penalty for diversion, or nothing. The penalty for exceeding the access time target consists of a fixed cost c_b and a cost c_e for each week that the treatment starts later than the target:

$$c(s,a) = \begin{cases} c_b + (a - b^r)c_e & \text{if } a - b^r > 0 \text{ (exceeding the access time target)}, \\ c_d & \text{if } a = 0 \text{ and request non-empty (diversion)}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

3.6. Optimality equations

Our objective is to find actions that minimize the total discounted expected costs over the infinite horizon. Therefore, the optimality equations are as follows:

$$v(s) = \min_{a \in A_s} \{c(s,a) + \lambda \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a) v(s')\} \qquad \forall s, \tag{6}$$

where $v(s)$ denotes the expected discounted cost associated with state s, s' denotes the next state where we transit to from state s, and λ is the discount factor. Note that the (immediate) costs of a decision are always incurred at the moment of the decision, so when the patient arrives. Future costs (which are discounted) are incurred on decisions about patients that might come in the future.

Term	Description
$d \in \mathcal{D}$	discipline
$w \in \mathcal{W}$	week in the booking horizon
$k(s) \in \mathcal{K}$	decision epoch in a week
$r(s) \in \mathcal{R}$	treatment request
$s, s' \in \mathcal{S}$	state, consisting of $r = \langle f^r, \ell^r, b^r \rangle$
$a \in \mathcal{A}_s$	feasible action, denoting the starting week of the treatment
$y_{dw}(s)$	no. of booked time slots for discipline d in week w
f_d^r	no. of consecutive weeks in which treatment from discipline d is required in request r
ℓ_d^r	no. of required time slots from discipline d in each treatment week of request r
q_d	total no. of available time slots per week for discipline d
b^r	access time target in request r
$c(s,a)$	costs, consist of penalty for exceeding access time target $c_b + (a - b^r)c_e$, diversion cost c_d, or nothing

Table 1: Indices and parameters in the MDP model.

4. Approximate dynamic programming solution approach

Even for very small instances, the size of the state space $((\prod_{d \in \mathcal{D}}(q_d + 1))^{(W-1)} * R * K)$ and the size of the action set (up to $W + 1$) of the MDP model prohibit finding a direct solution of its optimality equations (6) in reasonable time. Instead of solving the MDP model directly, we develop an approximate dynamic programming (ADP) approach, that enables us to approximate the solution of the MDP. In particular, we use the linear programming approach to ADP to obtain an approximate solution to the model and derive approximate optimal booking policies.

In order to deal with an intractable number of states, we first rewrite our MDP model in its equivalent linear programming form. The linear programming approach to discounted infinite-horizon MDPs is based on writing the optimality equations (6) as:

$$max_v \sum_{s \in \mathcal{S}} \alpha(s) v(s), \tag{7}$$

subject to

$$c(s,a) + \lambda \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a) v(s') \geq v(s) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}_s, \tag{8}$$

where α represents the weight of state s in the objective function. The solution $v(s)$ to (7-8) is equivalent to that of the optimality equations in (6) whenever α is strictly positive [24].

The model in (7-8) has one variable for every state s and one constraint for every feasible state-action pair (s,a), making its solution numerically intractable.

4.1. Value function approximation

A common approach in ADP is to derive approximate policies based on value function approximations. Value functions can be approximated by linear regression, where a (small) series of explanatory variables (or 'features') can be used to identify a set of basis functions $\phi_g(s)$ with weights Φ_g that represent the value function in the MDP formulation or, equivalently, the variables in the corresponding linear programming model: $v(s) = \sum_g \phi_g \Phi_g$ [34].

The choice of basis functions is not straightforward and can influence the results dramatically [22, 19]. To select a proper set of basis functions that together explain the value function to the largest extent, we use regression analysis of several basis functions on a small problem instance (see Appendix B). Based on this analysis, we use a value function approximation consisting of (i) $D \times W$ features of the type 'number of booked time slots for discipline d in week w', (ii) R features of the type 'current request type', (iii) one feature valuing 'the number of decision epochs still to go in the current week', and (iv) one feature denoting a constant. Together, these features explain a large part of the variance in the computed MDP state values ($R^2 = 0.75$). This entails the following affine approximation to $v(s)$:

$$v(s) = U_0 + \sum_d \sum_w y_{dw}(s) Y_{dw} + \sum_r e_r(s) E_r + \bar{k}(s) N \qquad \forall s, \tag{9}$$

$$U_0 \in \mathbb{R}, Y_{dw} \geq 0 \ \forall d, w, E_r \geq 0 \ \forall r, N \geq 0,$$

where $y_{dw}(s)$ follows from the state description; $e_r(s) = 1$ if the state description contains the r^{th} type of request and 0 otherwise; $\bar{k}(s) = K - k(s) + 1$ denotes the number of decision epochs still to go in the current week, and U_0 is a constant.

Now we can substitute the value function approximation (9) and the state transitions (1) and (3) into the linear program (7). For this, we use the following notation:

$$\gamma_{dw}(s,a) = y_{dw}(s) - \lambda y_{dw}(s'),$$
$$\epsilon_r(s,a) = \epsilon_r(s) = e_r(s) - \lambda e_r(s') = e_r(s) - \lambda \mathbb{P}(r),$$
$$\kappa(s,a) = \kappa(s) = \bar{k}(s) - \lambda \bar{k}(s') = \bar{k}(s) - \lambda(K - mod(k(s), K)),$$

Thus, rewriting the linear program (7), we obtain:

$$\max_{(U_0,Y,E,N)} \left(U_0 + \sum_d \sum_w \left[\sum_s \alpha(s) y_{dw}(s) \right] Y_{dw} + \sum_r [\sum_s \alpha(s) e_r(s)] E_r + [\sum_s \alpha(s) \bar{k}(s)] N \right), \quad (10)$$

subject to

$$(1-\lambda)U_0 + \sum_d \sum_w \gamma_{dw}(s,a) Y_{dw} + \sum_r \epsilon_r(s,a) E_r + \kappa(s,a) N \le c(s,a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}_s, \qquad (11)$$

$$U_0 \in \mathbb{R}, Y_{dw} \ge 0 \; \forall d, w, E_r \ge 0 \; \forall r, N \ge 0.$$

Note that, instead of $v(s)$, the decision variables are now the approximation parameters (U_0, Y, E, N).

In the MDP setting, the choice of $\alpha(s)$ in (7) does not influence the solution, as long as it is strictly positive [24]. This can be explained by the fact that for any state s and any feasible solution v we have $v^*(s) \ge v(s)$, in which v^* is the optimal solution. In the ADP setting, this phenomenon is lost and the states compete in importance, measured by their weights α. Intuitively, the weight of a state is therefore best chosen proportional to its frequency of occurrence over time. We introduce the following notation:

$$\mathbb{E}_\alpha[y_{dw}] = \sum_{s \in S} \alpha(s) y_{dw}(s) \quad \forall d, w,$$
$$\mathbb{E}_\alpha[e_r] = \sum_{s \in S} \alpha(s) e_r(s) \quad \forall r,$$
$$\mathbb{E}_\alpha[\bar{k}] = \sum_{s \in S} \alpha(s) \bar{k}(s),$$

which denote the expected values of $y_{dw}(s), e_r(s), \bar{k}(s)$ under the probability distribution α that reflects the occurrence of each state over time. Using this notation in the optimality equations (10), we obtain:

$$max_{(U_0,Y,E,N)} \left(U_0 + \sum_d \sum_w \mathbb{E}_\alpha[y_{dw}] Y_{dw} + \sum_r \mathbb{E}_\alpha[e_r] E_r + \mathbb{E}_\alpha[\bar{k}] N \right), \qquad (12)$$

subject to

$$(1-\lambda)U_0 + \sum_d \sum_w \gamma_{dw}(s,a)Y_{dw} + \sum_r \epsilon_r(s)E_r + \kappa(s)N \leq c(s,a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}_s, \tag{13}$$

$$U_0 \in \mathbb{R}, Y_{dw} \geq 0 \ \forall d, w, E_r \geq 0 \ \forall r, N \geq 0.$$

4.2. Column generation

The approximate linear programming model in (12) has a tractable number of variables, but still an intractable number of constraints. Therefore, we solve its dual (14) using column generation:

$$min_x \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} c(s,a)X(s,a), \tag{14}$$

subject to

$$(1-\lambda) \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} X(s,a) = 1,$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} \gamma_{dw}(s,a)X(s,a) \geq \mathbb{E}_\alpha[y_{dw}] \quad \forall d, w,$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} \epsilon_r(s)X(s,a) \geq \mathbb{E}_\alpha[e_r] \quad \forall r,$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} \kappa(s,a)X(s,a) \geq \mathbb{E}_\alpha[\bar{k}],$$

$$X(s,a) \geq 0 \quad \forall s \in \mathcal{S}, a \in \mathcal{A}_s.$$

The idea behind column generation is the following. Since most of the variables associated with state-action pairs will be non-basic and have value zero in the optimal solution, only a subset of the state-action pairs need to be considered when solving the dual optimization problem [8]. With column generation, we find the optimal solution to (14), the *master problem*, by starting with a small set of feasible state-action pairs and adding the state-action pair associated with the most violated primal constraint. Then the master problem is solved again, and this process is repeated until no primal constraint is violated anymore.

Finding an initial set of feasible state-action pairs focuses on dual feasibility, satisfying the constraints in (14), rather than optimality. Usually, an initial set can either be found manually [21], or, if this is not possible, using an optimization model [25]. In our case, an initial set can be found manually, starting by filling in the transition equations (1)-(4) in the constraints of (14). We use the following expected values in the right hand side of (14): $\mathbb{E}_\alpha[y_{dw}] = q(d) * \lambda^{(w-1)} \ \forall d, w < W$, $\mathbb{E}_\alpha[y_{dW}] = 0 \ \forall d$, $\mathbb{E}_\alpha[e_r] = \mathbb{P}(r)$ and $\mathbb{E}_\alpha[\bar{k}] = \frac{K+1}{2}$, which represent an initial situation where the first week is fully booked, and the number of booked slots decreases with the weeks until no slots are booked in the last week of the planning horizon. All decision epochs are equally probable and the requests are sampled according to their probability distribution.

For these expected values, we consider the following feasible state-action pair for each request type: the state s such that $y_{dw}(s) = q_d \ \forall d, w$; $\bar{k}(s) = K$; and the action $a = 0$. That is, all slots are fully booked, we are in the first decision epoch, and the action is to divert the treatment request. Together, these pairs form a feasible initial set of state-action pairs.

We identify the state-action pair associated with the most violated primal constraint (13) by solving the following optimization problem:

$$\operatorname{argmin}_{s\in S, a\in A_s}\left\{c(s,a) - [(1-\lambda)\tilde{U}_0 + \sum_d \sum_w \gamma_{dw}(s,a)\tilde{Y}_{dw} + \sum_r \epsilon_r(s)\tilde{E}_r + \kappa(s)\tilde{N}]\right\}, \qquad (15)$$

in which $(\tilde{U_0}, \tilde{Y}, \tilde{E}, \tilde{N})$ denote the optimal values of the dual LP (14) under the current set of state-action pairs. The state-action pair obtained in (15) enters the basis if:

$$\operatorname{argmin}_{s\in S, a\in A_s}\left\{c(s,a) - [(1-\lambda)\tilde{U}_0 + \sum_d \sum_w \gamma_{dw}(s,a)\tilde{Y}_{dw} + \sum_r \epsilon_r(s)\tilde{E}_r + \kappa(s)\tilde{N}]\right\} < 0. \qquad (16)$$

The column generation algorithm iterates until no primal constraint is violated, giving us the optimal values (U_0^*, Y^*, E^*, N^*). We can identify the approximate optimal policy $d^*(s)$, by inserting the values (U_0^*, Y^*, E^*, N^*) into the right hand side of the optimality equations (6):

$$d^*(s) \in \operatorname{argmin}_{a\in A(s)}\left\{c(s,a) + \lambda[U_0^* + \sum_d \sum_w y_{dw}(s')Y_{dw}^* + \sum_r e_r(s')E_r^* + \bar{k}(s')N^*]\right\} \qquad \forall s \in \mathcal{S}. \qquad (17)$$

Rearranging the terms, leaving out the ones that are independent of the action a, we obtain:

$$d^*(s) \in \operatorname{argmin}_{a\in A(s)}\{c(s,a) + \lambda \sum_d \sum_w y_{dw}(s')Y_{dw}^*\} \qquad \forall s \in \mathcal{S}. \qquad (18)$$

Expression (18) represents the balance between immediate costs $c(s,a)$ and the loss of available treatment capacity in the future $\sum_d \sum_w y_{dw}(s')Y_{dw}^*$. Filling in the definitions of $c(s,a)$ and $y_{dw}(s')$, and leaving out the terms independent of action a, we obtain for $k(s) < K$:

$$d^*(s) \in \operatorname{argmin}_{a\in A(s)}\begin{cases} c_b + (a-b^r)c_e + \lambda \sum_d \ell_d^r \sum_{w=a}^{a+f_d^r-1} Y_{dw} & \text{if } a - b^r > 0, \\ c_d & \text{if } a = 0 \text{ and request non-empty}, \\ \lambda \sum_d \ell_d^r \sum_{w=a}^{a+f_d^r-1} Y_{dw} & \text{otherwise.} \end{cases} \qquad (19)$$

For $k(s) = K$, we obtain:

$$d^*(s) \in \mathrm{argmin}_{a \in A(s)} \begin{cases} c_b + (a - b^r)c_e + \lambda \sum_d \ell_d^r \sum_{w=a-1}^{a+f_d^r-2} Y_{dw} & \text{if } a - b^r > 0, \\ c_d & \text{if } a = 0 \text{ and request non-empty}, \\ \lambda \sum_d \ell_d^r \sum_{w=a-1}^{a+f_d^r-2} Y_{dw} & \text{otherwise.} \end{cases} \tag{20}$$

From expressions (19) and (20), we see that the function on the right-hand side depends on the action a, the request $r(s)$, and the decision epoch $k(s)$. Expression (18) can therefore be written as a function of a, $r(s)$ and $k(s)$:

$$d^*(s) \in \mathrm{argmin}_{a \in A(s)} f(a, r(s), k(s)) \quad \forall s \in \mathcal{S}. \tag{21}$$

This observation will be used in Section 5 to provide insight into the resulting policies.

5. Numerical results

In this section, we present the results of the numerical experiments we carry out to evaluate the performance of the proposed solution approach. First, we compare the MDP and ADP models for a small size instance, followed by a case study based on data from the Sint Maartenskliniek. The MDP and ADP models are implemented in MATLAB R2014b. For the experiments, we used a 2.3 GHz Intel Core i5 Notebook with 8 GB RAM.

5.1. Small case - comparison of MDP and ADP model

We solve the MDP and ADP model with 3 weeks, 2 disciplines, 2 decision epochs per week, 2 time slots per week for both disciplines, and request types as indicated in Table 2. The diversion cost is 10,000 and the costs for exceeding the access time targets are 100 plus 25 per week. The discount factor is set to 0.98. This corresponds with a medium-term planning horizon that is applicable in this healthcare setting. By discounting slightly, the weight of the costs is relatively similar over the short-term while still less valued far in the future.

The output of both the MDP and the ADP model is a policy for every state. The average discounted cost of the system when following these policies, can be obtained from the MDP model directly.

The optimal approximation parameter values for the ADP model (U_0, Y, E, N) are given in Table 3. The Y^* values are slightly higher for the first week than for the other weeks. This indicates that the availability in this week contributes slightly more to the value of a state (i.e., is somewhat more important) than the availability in weeks later on in the planning horizon. The Y^* values are equal for both disciplines, since the demand and availability for both disciplines are

Request type r	$\mathbb{P}(r)$	b^r	Disc.1	Disc.2
			ℓ_d^r; f_d^r	ℓ_d^r; f_d^r
Urgent	0.3	1 week	2 slots; 1 week	2 slots; 1 week
Regular	0.7	2 weeks	2 slots; 1 week	2 slots; 1 week

Table 2: Parameter values for the requests of the small case.

Approximation parameter	Optimal value ($\times 10^5$)	
U_0	2.0338	
	Disc.1	*Disc.2*
Y_{d1} (week 1)	0.0253	0.0253
Y_{d2} (week 2)	0.0250	0.0250
Y_{d3} (week 3)	0.0247	0.0247
	Req.1	*Req.2*
E_r	0.1734	0.1734
N	0.0489	

Table 3: Values of the approximation parameters (U_0^*, Y^*, E^*, N^*) in the small case.

equivalent. For the E^* values, both requests contribute equally to the value of a state. Apparently, the different access time targets of both requests do not influence the E^* values in this case.

The values of the function $f(a, r(s), k(s))$ are displayed in Figure 2 for actions $a > 0$. For action $a = 0$, the value is $f(0, r(s), k(s)) = c_d = 10,000\ \forall r, k$. We can derive the policy for each state s by choosing the action with the smallest function value from the set of feasible actions. In Figure 2, we see, e.g., that regular requests in states with $k < K$ should be booked in week 2 (if possible), and otherwise in week 1 (if possible), or week 3. Diversion is advised only if there is not enough capacity available to start in weeks 1-3. Thus, for states with $k(s) < K$, urgent requests should be booked as soon as possible, and for regular requests, it is advised to book just within the access time target. For states with $k(s) = K$, no other requests can still arrive that can be booked in week 1. Therefore, it is best to book the current request as soon as possible, such that available capacity in week 1 is not left unused.

From the 2916 states in the small problem instance, there are eight states for which the policy generated by the ADP model differs from the MDP policy. In all these eight states, the approximate optimal policy prescribes action $a = 3$ for the urgent request where the MDP policy prescribes action $a = 2$. Both actions involve an exceedance of the access time target.

The average discounted cost of the system when following these policies, can be obtained from the MDP model directly by plugging in the corresponding policies. The average discounted costs over the infinite horizon are 254,943.714 for the MDP model and 254,943.728 for the ADP model (a difference of 5×10^{-6}%).

Considering the results in this section, we conclude that the model provides reasonable approximations of the optimal policy.

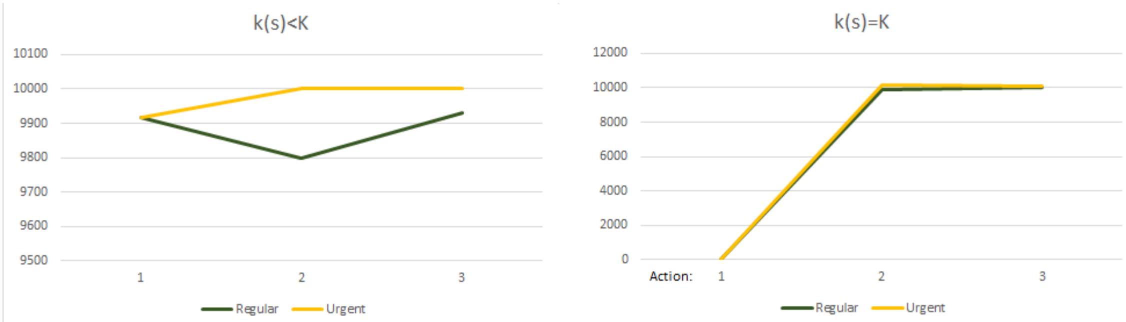

Figure 2: Values of the function $f(a, r(s), k(s))$ for $a > 0$ in the small case.

5.2. Case study at the Sint Maartenskliniek

We consider a case study based on data of the Spinal Cord Injury Unit of the Sint Maartenskliniek, with five disciplines and an arrival rate of 1.72 patients per week. According to Appendix A, 10 decision moments per week suffice in this case. Based on analysis of the historical data, we identified 13 request types (plus the empty request) as indicated in Table 4. The corresponding access time targets were determined using expert opinion from within the facility. The weekly total number of time slots per discipline can be found in Table 5. The costs for diversion are 10,000 and the costs for exceeding the access time target are 2000 plus 500 per week. This reflects the hospital's policy to only divert patients as a last resort, and the fixed exceeding costs entail an extra impulse to comply to the access time target. The planning horizon is set to 40 weeks, as for each request, the length of the treatment plus the access time target is at most 26 weeks and we do not want to book more than 14 weeks after the access time target. The discount factor is again set to 0.98. In this setting, the execution time of the ADP algorithm is 20 minutes (without the use of a time limit).

The optimal approximation parameter values are given in Table 6. With respect to the Y^* values, we see that only available slots of discipline 1 have a nonzero value. This indicates that the capacity of discipline 1 is the most important factor in the booking decision. The fact that discipline 1 is the bottleneck could also be seen from Table 5, where the system load is the highest for discipline 1. Analogous to the small case, the Y^* values are slightly higher in the first week than in the weeks later in the planning horizon. For the E^* values, we see that the shorter the access time target of the request and the heavier the treatment with respect to discipline 1, the higher the value of the request.

The values of $f(a, r(s), k(s))$ are given in Figure 3 for a subset of the requests. In this Figure, actions $a > 0$ are displayed. For $a = 0$, the value is $f(0, r(s), k(s) = c_d = 10,000\ \forall k$ for all shown requests. Instead of displaying all requests, we provide the results of two urgent requests (light color) and two regular requests (dark color). The requests consist of a resource intensive, long treatment (solid lines) or a shorter, less intensive treatment (dashed lines). Actions that imply treatment beyond the last week of the booking horizon are not shown. Again, we can derive the

Request type r	$\mathbb{P}(r)$	b^r	Disc.1 ℓ_d^r; f_d^r	Disc.2 ℓ_d^r; f_d^r	Disc.3 ℓ_d^r; f_d^r	Disc.4 ℓ_d^r; f_d^r	Disc.5 ℓ_d^r; f_d^r
1. Arm/hand screening	0.002	6 weeks	0;0	6; 1	4; 1	0;0	0;0
2. Baclofen Pump	0.009	8 weeks	0;0	0; 0	10;2	0;0	0;0
3. Bolus Baclofen	0.004	8 weeks	0;0	0; 0	8; 1	0;0	0;0
4. Decubitus	0.016	3 weeks	0;12	2; 12	6; 12	0;0	0;0
5. Guillain Barré	0.003	2 weeks	4;24	7; 24	21;24	4;24	4;24
6. Oncology	0.005	3 weeks	2;6	11;6	14;6	2;6	2;6
7. Sitting advice	0.016	6 weeks	0;0	6; 1	6; 1	0;0	0;0
8. Spinal cord T6 & higher	0.041	1 week	2;24	14;24	19;24	2;24	2;24
9. Spinal cord T7 & lower	0.033	1 week	2;18	8; 18	21;18	2;18	2;18
10. Spinal cord (others)	0.003	1 week	2;18	7; 18	18;18	2;18	2;18
11. Sports desk	0.001	1 week	4;1	0; 0	0; 0	4;1	4;1
12. Regular outpatient	0.034	4 weeks	2;18	6; 18	12;18	2;18	2;18
13. Continuation outpatient	0.005	1 week	2;12	8; 12	13;12	2;12	2;12
14. Empty request	0.828	0 weeks	0;0	0; 0	0; 0	0;0	0;0

Table 4: Parameter values for the requests of the case study, with ℓ_d^r in slots and f_d^r in weeks.

Discipline d	No. of slots q_d	System load
1. Movement therapy	90	0.55
2. Occupational therapy	557	0.44
3. Physiotherapy	898	0.49
4. Social work	175	0.24
5. Psychology	156	0.26

Table 5: Number of weekly available time slots per discipline in the case study and the system load per discipline in this case: average demand divided by capacity.

policy from the Figure. For states with $k(s) < K$, urgent requests should be booked as soon as possible, and for regular patients, it is advised to book just within the access time target. For each request, there is a point in time where $f(a, r(s), k(s))$ exceeds the diversion cost $c_d = 10,000$, i.e., from where diversion is preferred over booking. For states with $k(s) = K$, each request should be booked as soon as possible, but if this is not possible, the policy is again to book within the access time target.

Following the approach of [25], we translate these 'preferred order of actions' into a practical table that can be used by planners in practice (Table 7). For each request, the number 1 indicates the most preferred booking week in the planning horizon. If this action is infeasible, number 2 indicates the next preferred week, etc. As an example, a request of a Guillain Barré patient can best be booked starting in week 2, than in week 1, and if this is not possible, he/she should be diverted.

5.3. Insights into the approximate optimal policy

This section provides insights into the properties of the approximate optimal policy (AOP). We analyze the impact of different parameter settings on the values of $f(a, r(s), k(s))$. In Section 5.2, we already observed the following with respect to the various requests. First, the longer and/or more intensive the treatment, the higher the values of $f(a, r(s), k(s))$. Therefore, in the AOP,

Approximation parameter	Optimal value						
U_0 $(\times 10^3)$	-56.499						
	Disc.1	*Disc.2*	*Disc.3*	*Disc.4*	*Disc.5*		
Y_{d1} (week 1)	150.42	0	0	0	0		
Y_{d2} (week 2)	147.47	0	0	0	0		
Y_{d3} (week 3)	144.47	0	0	0	0		
Y_{d3} (week 4)	141.58	0	0	0	0		
Y_{d3} (week 5)	138.75	0	0	0	0		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		
Y_{d3} (week 20)	102.47	0	0	0	0		
Y_{d3} (week 21)	0	0	0	0	0		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		
Y_{d3} (week 40)	0	0	0	0	0		
	Req.1	*Req.2*	*Req.3*	*Req.4*	*Req.5*	*Req.6*	*Req.7*
E_r $(\times 10^3)$	0	0	0	0	10.000	1.683	0
	Req.8	*Req.9*	*Req.10*	*Req.11*	*Req.12*	*Req.13*	*Req.14*
	7.000	6.586	6.586	2.602	4.404	5.238	0
N $(\times 10^3)$	0.302						

Table 6: Values of the approximation parameters (U_0^*, Y^*, E^*, N^*) in the case study.

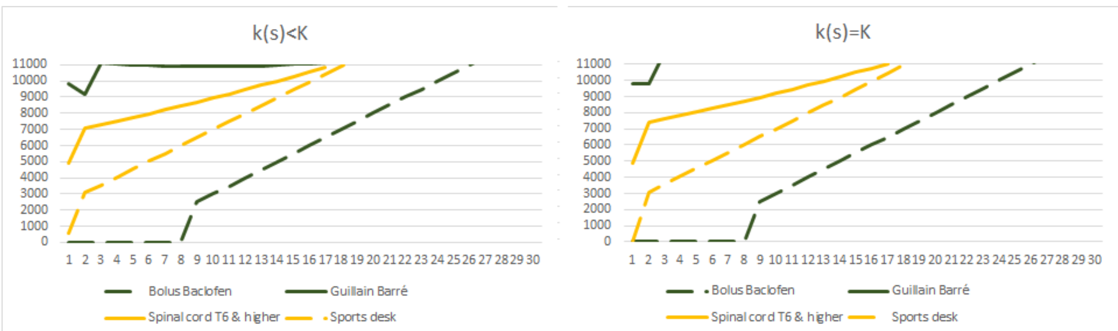

Figure 3: Function $f(a, r(s), k(s))$ for a subset of the requests in the case study.

Request type	Week in the booking horizon																
	1	2	3	4	5	6	7	8	9	...	16	17	...	23	24	...	40
Bolus Baclofen	8	7	6	5	4	3	2	1	9	...	16	17	...	23			
Guillain Barré	2	1															
Spinal cord T6 & higher	1	2	3	4													
Sports desk	1	2	3	4	5	6	7	8	9	...	16						

Table 7: Booking week preferences for $k(s) < K$, for a subset of the request types. For each request type, the number 1 indicates the most preferable booking week. If this action is infeasible, move to number 2, etc.

Discipline d	Scenario e		Scenario f	
	No. of slots q_d	System load	No. of slots q_d	System load
1. Movement therapy	75	0.66	75	0.66
2. Occupational therapy	393	0.62	366	0.68
3. Physiotherapy	669	0.65	656	0.71
4. Social work	123	0.35	65	0.63
5. Psychology	111	0.38	63	0.64

Table 8: Number of weekly available time slots per discipline in the case study and the system load in this case: average demand divided by capacity.

longer and/or more intensive treatments are more likely to be diverted. Second, the more capacity required of the most limited discipline(s), the higher the values of $f(a, r(s), k(s))$. This means that requests that require more capacity of a limited discipline are more likely to be diverted. Third, if the access time target is exceeded, the values of $f(a, r(s), k(s))$ have a steep increase over the weeks; before the target is reached, $f(a, r(s), k(s))$ is relatively similar over the weeks.

We now consider several changes to the parameters of the case study and show their influence on the AOP. Each scenario assumes the settings from Section 5.2, referred to as 'base case', with one change at a time. The individual changes and their main results are described below. In Figure 4, the values of $f(a, r(s), k(s))$ are displayed for $k < K$ against the base case.

a) Discount factor: when increasing the discount factor to 0.99, we observe higher values for in particular the longer/more intensive treatments. Therefore, policies for longer/more intensive treatments make an earlier switch to diversion.

b) Costs: when increasing the costs for exceeding the access time target to 3000 plus 1000 per week, while the costs for diversion stay 10,000, we observe steeper values for all cases and therefore, policies switch to diversion earlier on.

c) Booking horizon: a smaller number of weeks W causes an earlier switch to diversion, as the opportunities to book capacity within the booking horizon are limited.

d) Patient mix: adapting the patient arrival rates such that 24% of the (non-empty) requests are urgent, while the number of patient arrivals stays the same, does not cause any change in the policy.

e) System load: when reducing the capacities of all disciplines to the numbers provided in Table 8, we observe higher values for in particular the longer/more intensive treatments. Therefore, policies for longer/more intensive treatments make an earlier switch to diversion.

f) Capacities: when changing the capacities of the disciplines such that not just one discipline is a bottleneck but multiple (see Table 8), the approximate parameter values of the first weeks change to those displayed in Table 9. The values of $f(a, r(s), k(s))$ change depending on the intensity of the treatments with respect to those new bottleneck disciplines.

To conclude, the approximate optimal policy always consists of booking requests just within their access time targets and divert them if they cannot be booked within certain intervals in the booking

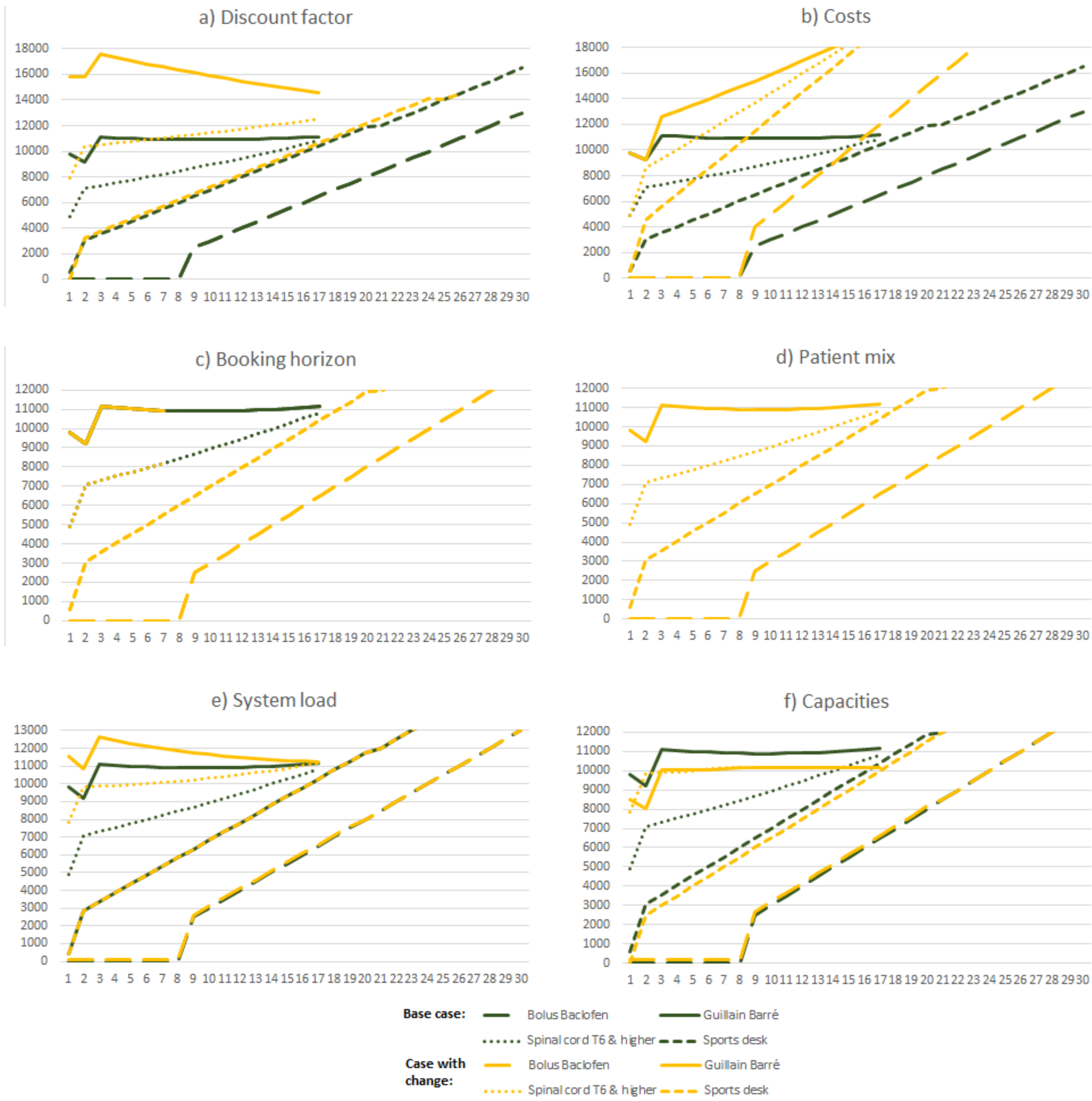

Figure 4: Function $f(a, r(s), k(s))$ for different settings of the case study.

Approximation parameter	Optimal value						
U_0 ($\times 10^3$)	-74.488						
	Disc.1	*Disc.2*	*Disc.3*	*Disc.4*	*Disc.5*		
Y_{d1} (week 1)	0	0.964	24.624	0	0		
Y_{d2} (week 2)	0	0.945	24.132	0	0		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		
	Req.1	*Req.2*	*Req.3*	*Req.4*	*Req.5*	*Req.6*	*Req.7*
E_r ($\times 10^3$)	0.096	5.528	0.174	1.578	8.706	1.987	8.859
	Req.8	*Req.9*	*Req.10*	*Req.11*	*Req.12*	*Req.13*	*Req.14*
	10.000	10.000	0.141	2.000	4.410	0.431	0
N ($\times 10^3$)	0.403						

Table 9: Values of the approximation parameters (U_0^*, Y^*, E^*, N^*) in the case study with scenario (f).

horizon. These intervals depend on the discount factor, the cost structure, the booking horizon, the system load, and the capacities of the most constrained discipline(s).

5.4. Comparison of the approximate optimal policy with the currently used booking policy

To evaluate the performance of the approximate optimal policy (AOP) obtained by the ADP model, we use discrete event simulation. Results are compared to the following three policies. First, the policy that resembles the hospital's current practice the most: book every request in the earliest week with enough available capacity, and if it is not possible to book the request within the booking horizon W, divert. We refer to this policy as the myopic policy. Second, the approximate optimal policy where diversion is only advised if it is not possible to book the request within the booking horizon W. We refer to this policy as the AOP without diversion. Third, a policy based on general structures of the policies found in [21, 25], applied to the online setting. We use parameters that seem reasonable for this case study. Requests with targets 1-3 weeks are booked as soon as possible and at most four weeks after their target; the other requests are booked from week 4, and at most four weeks after their target. Diversion is advised if there is not enough capacity available in the described intervals. We refer to this policy as the heuristic policy.

We evaluate three scenarios: the case study at the SMK, the same case study with a different case mix, and the case study at the SMK with a higher system load. The latter two scenarios and their policies are described in Section 5.3, changes (d) and (e).

The length of a simulation run is set to 500 weeks and statistics are collected for 140 runs, starting from different states with respect to $r(s)$ and $k(s)$. The booked capacity in the initial states is $y_{dw}(s) = q(d) * \lambda^{(w-1)}\ \forall d, w < W, s$ and $y_{dW}(s) = 0\ \forall d, s$. The runs are repeated 10 times with different arrival patterns. We use common random numbers to compare the policies. The simulation is implemented in MATLAB R2014b and the execution time in this setting is 20 minutes.

The expected discounted cost of the system, as well as the percentage of diversions and the percentage of requests booked after the access time targets, are given in Tables 10, 11 and 12 for each of the scenarios respectively. A subdivision of the percentages is given for 'urgent' requests (with an access time target of 1 week) and for 'regular' requests (with an access time target longer than 1 week).

In the first scenario, the percentage of requests booked after their access time target is relatively high for all policies, and in particular for urgent patients. Since urgent requests (which account for 48% of the arrivals) have relatively resource intensive and long treatments (representing 79.7% of the required capacity), which are more difficult to fit in an already filled schedule than shorter, less intensive treatments. It has to be noted that in reality, the management of the Sint Maartenskliniek would always try to avoid booking requests after their targets, by giving patients slightly more or less therapy than prescribed or by planning overtime for bottleneck disciplines. Comparing the AOP to the other policies, the AOP performs best, and the largest improvement in the percentage of requests booked within their access target can be observed for urgent patients.

In the second scenario, where 24% of the (non-empty) requests are urgent, the percentages of requests booked within their access time target are much higher than in the first scenario. Although the system load is similar to the load in scenario 1, the lower percentage of urgent requests clearly has a good influence on the percentages of requests booked within their target. In this scenario, the AOP still outperforms the other policies, but the differences between the AOP, the AOP without diversions and the heuristic policy are small. Apparently, in this scenario, the advise to divert requests if they cannot be booked within a certain interval in the booking horizon is not necessary to reserve space for future urgent requests.

The third scenario, where the capacities of the disciplines are reduced such that the system has a higher load, shows a much better performance of the AOP. We conclude that in this scenario, the advise to divert requests from some point in the booking horizon is a valuable ingredient of the AOP, to ensure enough space is kept free for future urgent requests.

Considering the simulation results, we conclude that the approximate optimal policy outperforms the other policies with respect to the percentage of requests booked within their access time targets. Thus, the policy of booking regular requests just within their access time targets and diverting them if they cannot be booked within a certain interval in the booking horizon, ensures more space for future urgent requests to be booked within their access time targets.

6. Conclusion and discussion

In this paper, we presented a methodology for online capacity planning of rehabilitation treatments. The method is designed in such a way that compliance to access time targets is optimized for all patients. The model considers online decision making regarding multi-priority, multi-appointment, and multi-resource capacity allocation. We formulated this problem as a Markov decision process (MDP), and developed an approximate dynamic programming algorithm to obtain approximate

	AOP	Myopic	AOP w.d.	Heuristic
Requests booked after target	21.5%	26.4%	25.0%	25.2%
Booked after target (urgent)	42.6%	53.4%	51.0%	51.2%
Booked after target (regular)	0.0%	2.3%	2.2%	2.2%
Diversions	0.2%	0.0%	0.0%	0.1%
Diversions (urgent)	0.0%	0.0%	0.0%	0.2%
Diversions (regular)	2.2%	0.0%	0.0%	0.0%
Requests booked within target	78.3%	73.6%	75.0%	74.7%
Booked within target (urgent)	57.4%	46.6%	49.0%	48.6%
Booked within target (regular)	97.8%	97.7%	97.8%	97.8%

Table 10: Results for scenario 1, using the approximate optimal policy (AOP), the myopic policy, the AOP without diversion and the heuristic policy.

	AOP	Myopic	AOP w.d.	Heuristic
Requests booked after target	3.6%	6.2%	3.8%	4.3%
Booked after target (urgent)	15.0%	24.0%	15.0%	17.0%
Booked after target (regular)	0.0%	0.2%	0.2%	0.2%
Diversions	0.1%	0.0%	0.0%	0.0%
Diversions (urgent)	0.0%	0.0%	0.0%	0.0%
Diversions (regular)	0.1%	0.0%	0.0%	0.0%
Requests booked within target	96.3%	93.8%	96.2%	95.7%
Booked within target (urgent)	85.0%	76.0%	85.0%	83.0%
Booked within target (regular)	99.9%	99.8%	99.8%	99.8%

Table 11: Results for scenario 2, using the approximate optimal policy (AOP), the myopic policy, the AOP without diversion and the heuristic policy.

	AOP	Myopic	AOP w.d.	Heuristic
Requests booked after target	8.3%	35.1%	30.8%	28.8%
Booked after target (urgent)	7.8%	55.9%	47.9%	45.6%
Booked after target (regular)	7.4%	10.9%	9.9%	9.5%
Diversions	11.1%	0.0%	0.0%	2.9%
Diversions (urgent)	16.9%	0.0%	0.0%	4.6%
Diversions (regular)	2.3%	0.0%	0.0%	0.0%
Requests booked within target	80.6%	64.9%	69.2%	68.3%
Booked within target (urgent)	75.3%	44.1%	52.1%	49.8%
Booked within target (regular)	90.3%	89.1%	90.1%	90.5%

Table 12: Results for scenario 3, using the approximate optimal policy (AOP), the myopic policy, the AOP without diversion and the heuristic policy.

solutions. Starting from a linear programming approach to ADP, we applied value function approximation and column generation to derive effective booking policies for online capacity planning of rehabilitation treatments. For the approximation of the value function, we provided a founded choice of basis functions. We provided insights into the characteristics of the optimal policy and evaluated the performance of the optimal policy using simulation.

The performance of the proposed methodology is evaluated using a case study based on data of the Sint Maartenskliniek. The results show that the approximate optimal policy outperforms the booking policy currently used in practice. In the simulation, the percentage of requests booked within their access time target increases from 73.6% to 78.3%, with the relatively largest increase for urgent patients (from 46.6% to 57.4%). The approximate optimal policy prescribes to book requests just within their access time targets and divert them if they cannot be booked within a certain interval in the booking horizon, where the interval depends on the capacities of the most constrained discipline(s). This policy ensures more space for future urgent patients to be booked within their access time targets.

It is important to note that the case study presented in this paper does not incorporate all details of the planning practices at the Sint Maartenskliniek. In our study, we use aggregated data of the treatment requests and the capacity of disciplines, which do not take all characteristics of the daily operations into account. As an example, the capacity of disciplines might change over the weeks and treatment requirements of a patient can change over the weeks, dependent on the patient's progress.

A possible extension to the proposed capacity planning method is to investigate the use of a threshold policy that tries to reserve some capacity for future high priority arrivals. Another direction for further research is to extend the planning decision by also determining the exact days, time slots and therapists for each request. Given the constraints and preferences that apply to this more detailed level of scheduling, and the increasing complexity of the state and action space if these decisions would be included, this is an interesting direction for further research. Besides the use of the approximate optimal policy, further improvement is possible in the compliance with the access times targets, especially for urgent patients. Suggestions for future research are to investigate the allowance for rescheduling capacity of already planned patients and for working in overtime. Also, planning in an offline fashion (collecting requests and executing them periodically, e.g., once per week) or planning using a waiting list, will most likely lead to better results in terms of access time compliance.

Given the results of the case study, we are convinced that the application of our method can be valuable to many rehabilitation facilities. The case of the Spinal Cord Injury Unit is representative for other rehabilitation facilities treating both urgent and regular patients. Furthermore, since many other multi-disciplinary care systems deal with booking capacity in an online fashion, many relevant possibilities exist for our method to be applied in other health care systems, e.g., in cancer treatment or rapid cancer diagnostics.

Acknowledgement

The authors would like to thank Hans Reijnierse for his useful comments and the personnel of the Rehabilitation Department of the Sint Maartenskliniek in Nijmegen for their involvement in the development of the proposed capacity planning method. This research has been supported by the Sint Maartenskliniek.

References

[1] Revalidatie Nederland (rehabilitation sector association The Netherlands), retrieved December 20, 2017. Available from www.revalidatie.nl/revalideren/home-r.

[2] R.J. Boucherie and N.M. van Dijk, editors. *International Series in Operations Research and Management Science: Markov decision processes in Practice*. Springer, 2017.

[3] A. Braaksma, N. Kortbeek, G.F. Post, and F. Nollet. Integral multidisciplinary rehabilitation treatment planning. *Operations Research for Health Care*, 3(3):145–159, 2014.

[4] T. Cayirli and E. Veral. Outpatient scheduling in health care: a review of literature. *Production and Operations Management*, 12(4):519–549, 2003.

[5] W. Dafoe, H. Arthur, H. Stokes, L. Morrin, and L. Beaton. Universal access: but when? Treating the right patient at the right time: access to cardiac rehabilitation. *Canadian Journal of Cardiology*, 22:905–911, 2006.

[6] D.P. De Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[7] D.P. De Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of operations research*, 29(3):462–478, 2004.

[8] G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors. *Column generation (Vol. 5)*. Springer Science Business Media, 2006.

[9] A. Erdelyi and H. Topaloglu. Approximate dynamic programming for dynamic capacity allocation with multiple priority levels. *IIE Transactions*, 43:129–142, 2010.

[10] J. Fan and R. Li. Statistical challenges with high dimensionality: Feature selection in knowledge discovery. *arXiv preprint math/0602133*, 2006.

[11] Y. Gocgun and A. Ghate. Lagrangian relaxation and constraint generation for allocation and advanced scheduling. *Computers & Operations Research*, 39:2323–2336, 2012.

[12] Y. Gocgun and M.L. Puterman. Dynamic scheduling with due dates and time windows: an application to chemotherapy patient appointment booking. *Health Care Management Science*, 17:60–76, 2014.

[13] M. Grötschel and O. Holland. Solution of large-scale symmetric travelling salesman problems. *Mathematical Programming*, 51:141–202, 1991.

[14] C. Guestrin, D. Koller, and R. Parr. Max-norm projections for factored MDPs. *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 673–682, 2001.

[15] D. Gupta and B. Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE transactions*, 40(9):800–819, 2008.

[16] P.J.H. Hulshof, M.R.K. Mes, R.J. Boucherie, and E.W. Hans. Patient admission planning using approximate dynamic programming. *Flexible Services and Manufacturingl*, 28:30–61, 2016.

[17] A.G. Leeftink, I.A. Bikker, I.M.H. Vliegen, and R.J. Boucherie. Multi-disciplinary planning in healthcare: a review. 2017.

[18] M.S. Maxwell, M. Restrepo, S.G. Henderson, and H. Topaloglu. Approximate dynamic programming for ambulance redeployment. *INFORMS Journal on Computing*, 22:266–281, 2010.

[19] M. Mes and A. Pérez Rivera. Approximate dynamic programming by practical examples. In R.J. Boucherie and N.M. van Dijk, editors, *Markov Decision Processes in Practice*, chapter 3, pages 61–101. Springer, 2017.

[20] J. Patrick and M.L. Puterman. Improving resource utilization for diagnostic services through flexible inpatient scheduling: A method for improving resource utilization. *Journal of the Operational Research Society*, 58:235–245, 2007.

[21] J. Patrick, M.L. Puterman, and M. Queyranne. Dynamic multipriority patient scheduling for a diagnostic resource. *Operations Research*, 56:1507–1525, 2008.

[22] W.B. Powell. *Approximate Dynamic Programming: solving the curses of dimensionality.* Wiley Series in Probability and Statistics, 2nd edition, 2011.

[23] K. Pruhs, J. Sgall, and E. Torng. Online scheduling. In J. Y.-T. Leung, editor, *Handbook of Scheduling: algorithms, models, and performance analysis*. CRC Press, Boca Raton, FL, 2004.

[24] M.L. Puterman. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, 2014.

[25] A. Sauré, J. Patrick, S. Tyldesley, and M.L. Puterman. Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223:573–584, 2012.

[26] J. Scheer, T. Kroll, M.T. Neri, and P. Beatty. Access barriers for persons with disabilities. *Journal of Disability Policy Studies*, 13:221–230, 2003.

[27] K. Schimmelpfeng, S. Helber, and S. Kasper. Decision support for rehabilitation hospital scheduling. *OR spectrum*, 34:461–489, 2012.

[28] V. Schmid. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research*, 219:611–621, 2012.

[29] H.J. Schütz and R. Kolisch. Approximate dynamic programming for capacity allocation in the service industry. *European Journal of Operational Research*, 218:239–250, 2012.

[30] H.J. Schütz and R. Kolisch. Capacity allocation for demand of different customer-product-combinations with cancellations, no-shows, and overbooking when there is a sequential delivery of service. *Annals of operations research*, 206:401–423, 2013.

[31] J. Sgall. On-line scheduling. In A. Fiat and G. Woeginger, editors, *Online Algorithms, vol. 1442 of Lecture Notes in Computer Science*, pages 196–231. Springer Berlin Heidelberg, 1998.

[32] H. Simao and W.B. Powell. Approximate dynamic programming for management of high-value spare parts. *Journal of Manufacturing Technology Management*, 20:147–160, 2009.

[33] H. Topaloglu and W.B. Powell. Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS Journal on Computing*, 18:31–42, 2006.

[34] J.N. Tsitsiklis and B. van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1-3):59–94, 1996.

Appendix A

We assume that new patients arrive according to a Poisson process with a rate of μ arrivals per week. The random number N of new patients who arrive within a typical decision epoch of length $1/K$ (in weeks) is then Poisson distributed with scale parameter μ/K. We have $\mathbb{P}(N = 0) = e^{-\mu/K}$, $\mathbb{P}(N = 1) = \frac{\mu}{K}e^{-\mu/K}$, and $\delta = \mathbb{P}(N \geq 2) = (1 - (1 + \frac{\mu}{K})e^{-\mu/K})$. In the case study of Section 5.2, the arrival rate $\mu = 1.72$. For $K = 10$, we find $\delta = 1.3\%$.

Appendix B

In Section 4.1, we introduce basis functions into the formulation to approximate the future value of an action a in a state s. The challenge is to make sure the choice of basis functions contributes to the quality of the solution. The basis functions can be observed as independent variables in the regression literature [22]. Hence, to select a proper set of basis functions that have significant impact on the value function, we use a regression analysis. In this analysis, the dependent variables are the optimal values from the MDP solution, and the independent variables are the basis functions calculated from the state descriptions.

Table 13 shows the regression results for various sets of basis functions. In each set, a constant is added as an explanatory variable. The R^2 depicts the variation in the value that is explained by the corresponding regression model.

Features	R^2	No. of variables
Combination of no. of booked slots for discipline d in week w, type of request r and no. of epochs to go in current week $\bar{k}$	0.752	$D \times W + R + 1$
Type of request r	0.610	R
No. of booked slots for discipline d in week w	0.090	$D \times W$
No. of booked slots for discipline d	0.087	D
No. of decision epochs to go in current week $\bar{k}$	0.052	1
Maximum no. of booked slots in week w	0.046	W
Maximum no. of booked slots for discipline d	0.014	D
No. of booked slots in each week w	0.009	W

Table 13: The different sets of basis functions and the R^2 regression values.

One can observe that the features with high level of detail about the state description score significantly better (are higher in the ordered table). The higher R^2, the more suitable the basis functions are for predicting the value. However, we have to take into consideration that a correct ranking of the different actions is more important than how well the value is predicted [19]. Therefore, policies obtained using a particular set basis functions, should be carefully evaluated before one can conclude that this set of basis functions is a good choice.

For our ADP model, we choose to use a value function approximation consisting of $D \times W$ features of the type 'number of booked time slots for discipline d in week w', R features of the type 'current request type', one feature valuing 'the number of decision epochs still to go in the current week', and one feature denoting a constant. These basis function explain a large part of the variability in the optimal values using the MDP model ($R^2 = 0.75$) and the obtained policy performs well (see Section 5.1). Another advantage is that the basis functions can be obtained directly from the state description.