Planning patients with preference in an outpatient department

Research Paper Business Analytics

Author: Anne Zuiker
Supervisor: René Bekker
5-10-2014

Faculty of Sciences
VU University
De Boelelaan 1081
1081 HV Amsterdam
Nederland

Preface

Part of the Master's programme in Business Analytics is writing a research paper. This paper is the result of an individual research combining the fields of business administration, mathematics and computer science. The goal of writing this research paper is to show my ability to describe a problem clearly to an expert manager and analyze it. Hopefully, after reading this paper you are convinced.

The problem we have investigated is about giving a patient more influence in scheduling his appointment in an outpatient department, while remaining an efficient schedule. With this report we present the results of investigating this problem.

I want to thank my supervisor René Bekker, assistant professor at VU University, for guiding me throughout this research. He always made time for me, gave constructive criticism and helpful suggestions. That helped me a lot throughout the research.

Anne Zuiker

October 2014

Summary

Recently, the focus on patient-focused care has increased severely in the health care domain. To keep up with the competition it is important for hospitals to satisfy the patients needs and preferences. An example of patient-focused care is letting the patient schedule his own appointments for an outpatient visit. But do you open up the complete schedule for the patient, where the patient will plan as close as possible to his preference, or do you give him a selection of the schedule based on his preference? This leads to the following research question:

- What is the best way of assigning slots to patients with preferences in an outpatient department?

Leonié Ottens [1] performed a study in this area and the main conclusion was that the dynamic programming (DP) model gave better results than the greedy model. However, the calculation time of the DP model increases strongly as the number of slots in a schedule grows. The calculation time is too large to use this model in an outpatient department. The aim of this research is therefore to extend the model of Ottens with new patients and a stochastic process in which the patients arrive, and to solve the problem with the increasing calculation time. With the new stochastic way of patient arrivals you do not know upfront how many patients you have to schedule.

We developed an improved DP model and this model includes two types of patients: the first type needs one slot because he comes for a follow up appointment. The second type needs two slots because it is a new patient. The DP model always gives lower values than the greedy model. But as there are more patients of type 2, this difference becomes bigger. Then the costs of the DP model are 24.5% smaller than the costs of the greedy model. Also, when the refusal costs are higher, this difference becomes bigger. Then the costs of the DP model are 46.8% smaller. So it depends on what the distribution is between the two types of patients and how high the refusal costs are, to see how much better the DP model actually is. But it is clear that the new way with stochastic arrivals of patients increases the profit you can get by using the DP model.

To solve the problem of the increasing calculation time, we developed a reduced DP model which uses clusters instead of slots. This way we have to keep track of less variables, which solves the problem. Unfortunately, the value function of the reduced DP model is an estimation and gives higher values than the greedy and the DP model. This means that the reduced DP model results in higher costs. We expect that the actual values of the reduced DP model are lower than they are now, because in the value function we add up two different types of costs instead of one. Further research is required to determine whether this is really the case.

We recommend to keep researching this problem, since the profit you can gain by using the DP model could be around 47%. We advise to compute the actual costs of the reduced DP model and to extend the model with multiple days. To approach reality even more, we would also recommend to take costs into account of being left over with consecutive empty slots in the middle of the schedule. The profit that can be achieved with further research will be worth it.

Table of contents

1. Introduction

Nowadays patients are becoming more conscious about where they can get the best care. Consequently, the competition in care becomes bigger and for hospitals it becomes more important to satisfy the patient's needs and preferences. An example of satisfying the patient's preferences is to give them a time and date for an appointment that they prefer. Most outpatient departments give each patient a strict time and date if they want to make an appointment. If the patient is not available at that time or that day, the appointment can be rescheduled, but this will cost extra effort from both the patient and the hospital. Therefore, it is desirable to let patients schedule their appointments instead of the hospital. But do you open up the complete schedule for the patient, where slots are taken in a greedy fashion, or do you give a selection of slots to the patient, depending on his preference? And if you give a selection, what selection should that be?

Appointment scheduling of patients with preference is a recent development in health care and therefore there is not much literature about this subject. The literature that can be found is referred to in the report of Léonie Ottens [1]. Ottens investigated appointment scheduling with patient preferences by using a technique called dynamic programming. Her model gave better results than using a greedy algorithm. However, the calculation time increased exponentially as the number of appointments increased because the state space becomes too large. Also, the model assumed that there was only one type of appointment and that each time step a patient wanted to book an appointment. So this research had its limitations and therefore could not be used directly in an outpatient department. A new model which takes new patients with high refusal costs into account would probably try to shield these new patients. Since the new patients are very important to a hospital, it would be beneficial to use this new model.

The aim of this research is to extend the model of Ottens and to find ways to deal with the exponential increasing calculation time. The model of Ottens will be extended by adding another patient type that needs a double appointment, making the process of planning patients stochastic and implement the possibility to refuse patients. This leads to the following research question:

- What is the best way of assigning slots to patients with preferences in an outpatient department?

To answer this question the following subquestions are addressed:

1. How much better does dynamic programming performs than the greedy algorithm for appointment scheduling?
2. How does the planning of multiple types of patients influence the way of planning?
3. How does the possibility of refusing patients influence the way of planning?
4. How can the problem of the exponential increasing calculation time be dealt with the use of dynamic programming?

In this research paper three scenarios are considered: the greedy policy, the dynamic programming model and the reduced dynamic programming model. The greedy policy gives each patient the nearest available slot(s) he/she prefers. The dynamic programming model calculates all possibilities and determines which is the best one at each time step and for each patient. The reduced dynamic programming model works with a reduced state space where less calculations are performed.

This paper is divided in four chapters. Chapter 2 describes the methods we have used throughout this research. This includes a description of the greedy model, the dynamic programming model and the reduced dynamic programming model, including pseudo code. Chapter 3 contains the validation of the greedy and dynamic programming model and the results following from the three models. Chapter 4 presents the conclusion and recommendations. Here we answer the research question and subquestions and give some recommendations for future research.

2. Methods

This chapter describes the model in paragraph 2.1 and the three models we have used in this research in section 2.2-2.4. The three models are the greedy model, the dynamic programming model and the reduced dynamic programming model. We explain the problem, how the models work, how the models are implemented with pseudo code and give an example. All models are programmed with JAVA in Eclipse.

2.1 Model description

We consider the problem for a fixed day: each day there is a schedule with N available slots. In each slot an appointment can be booked. There are M time steps where an event can happen. There are three types of events:

- type 0: no patient arrives that wants to make an appointment.
- type 1: a patient arrives that wants to make an appointment for one slot. This is a patient that has been to the hospital before and needs a checkup.
- type 2: a patient arrives that wants to make a appointment for two consecutive slots. This is a new patient and therefore the patient needs a double appointment.

Each type of event has a probability of occurrence (p_0, p_1, p_2), where the sum of the three probabilities is one. Thus each time step an event happens: a dummy event where no patient arrives, or a patient of type 1 or 2 arrives. Each event is independent of the previous event. The expected number of occupied slots $\rho(M)$ of the schedule after M steps is as follows:

$$\rho(M) = M * (p_0 * 0 + p_1 * 1 + p_2 * 2). \tag{2.1}$$

For each different size of schedule, the number of time steps is chosen such that the schedule is expected to be 95% full. If the schedule is full, or the patient cannot be scheduled for that day, it will be refused. This means that his appointment will be moved to the next day. It may happen that a patient is refused even though the schedule is not full, because it is better to keep slots empty for patients that arrive later. This may be the case when the free slots are far apart from the preferred slot. Examples of this situation will be discussed in chapter 3.

Each patient arrives with a preference for a certain slot. Ottens [1] has proposed three different formulas for the distribution of the preference for some slot. The first formula describes the situation where the probability of a preference for each slot is equal:

$$g_1(x) = \frac{1}{N}. \tag{2.2}$$

The second formula entails that patients are more likely to prefer an outer slot of the schedule. So this means that patients prefer the slots at the beginning and at the end of the day.

$$g_2(x) = \frac{\left(x-\frac{1}{N}*\sum_{n=1}^{N} n\right)^2+1}{\sum_{x=1}^{N}\left(\left(x-\frac{1}{N}*\sum_{n=1}^{N} n\right)^2+1\right)}. \quad (2.3)$$

The last formula describes the opposite situation, where patients prefer the slots in the middle of the day:

$$g_3(x) = \frac{-\left(\left(x-\frac{1}{N}*\sum_{n=1}^{N} n\right)^2\right)+\max_{x\in\{1,\dots,N\}}\left(x-\frac{1}{N}*\sum_{n=1}^{N} n\right)^2}{\sum_{x=1}^{N}\left|-\left(\left(x-\frac{1}{N}*\sum_{n=1}^{N} n\right)^2\right)+\max_{x\in\{1,\dots,N\}}\left(x-\frac{1}{N}*\sum_{n=1}^{N} n\right)^2\right|}. \quad (2.4)$$

The three preference functions are used to compute the probabilities of a preference for slot k for type 1 and type 2: p_k^1 and p_k^2. This entails that $p_k^i = g_j(k)$, where $j\epsilon\{1,2,3\}$ and $i\epsilon\{1,2\}$. For type 2 the preference function is different. Since the patient wants an appointment for two slots, he can have four preferences in a schedule with five slots: for slot 1 + 2, slot 2 + 3, slot 3 + 4 and slot 4 + 5. Therefore, in the functions N is substituted with $N-1$.

Once a patient is scheduled for a slot and this is not the slot the patient preferred, costs are involved. Ottens [1] also proposed three different formulas for the distribution of the costs, where l is the assigned slot and k the preferred slot. Figure 1 shows the behavior of these three cost function. The first formula is called the linear cost function. This formula implies that once the distance between the preferred and assigned slot increases, the costs increases linearly.

$$f_1(l,k) = \frac{1}{N} * \sqrt{(l-k)^2}. \quad (2.5)$$

The second formula is the quadratic cost function. Here the costs increase quadratic, once the distance between the preferred and assigned slot increases. This means that the costs first increase slowly, but after a certain point start to increase rapidly.

$$f_2(l,k) = \frac{1}{N^2} * (l-k)^2. \quad (2.6)$$

Third is the saturating cost function. As you can see in figure 1, the costs first increase slowly, then faster in the middle and slower at the end. Thus at a certain distance between the preferred and assigned slot, the costs start to saturate.

$$f_3(l,k) = 1 - \exp\left(-\frac{1}{2*4^{\frac{N}{5}}} * (l-k)^2\right). \quad (2.7)$$

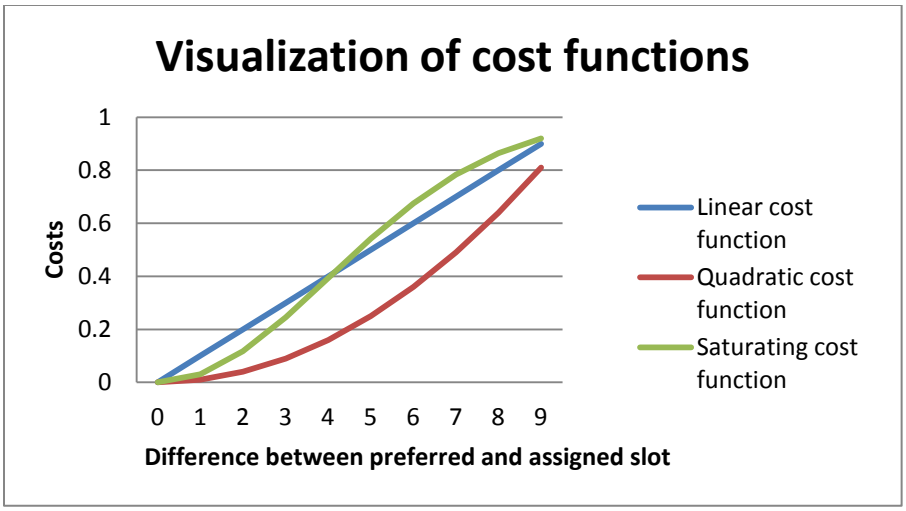


Figure 1: visualization of cost functions

In the program we use a binary representation for the occupied slots in the schedule as an index. We use this representation for the greedy and the DP model. Consider the following schedule with five slots:

1	2	3	4	5

At index zero, there are no occupied slots. At index one only slot 5 is occupied (00001) and at index two slot 4 is occupied (00010). At index three both slot 4 and slot 5 are occupied (00011). The complete list with the index and corresponding schedules according to the binary representation can be found in appendix B.

2.2 Greedy model

The greedy model is the simplest model and will be used to compare with the results of the dynamic programming model. Each time step the greedy policy assigns the slot(s) that the patient prefers, or the free slot(s) that is/are closest to the preferred slot(s), with a preference for the left slot. For example, if we have the following schedule, with black slots representing booked slots:

1	2	3	4	5

If a patient of type 1 arrives and prefers the second slot, the greedy policy will assign the first slot, because the closest available slots are 1 and 3 and the model has a preference for left slots. If the second and third slots are both not available and an arriving patient prefers the third slot, the fourth slot will be assigned. Once the second and fourth slots are assigned, it is not possible to plan a patient of type 2. This suggests that the greedy model is not optimal and can be improved. But how can we compare two different models? We have to calculate a value $V_j(J)$ which is the value of the model at time step j for schedule J. The value gives an impression of how high the costs are. So a lower value means that the costs for this model will be lower and therefore the model is better.

Ottens [1] has derived four formulas that are used to calculate the value function of a model. We have extended these formulas to take refusals into account. For all three models we start in the last time step where no events can occur anymore and go back to the first time step where the schedule is empty. This process comes from the dynamic programming model, but here we assign a slot according to the greedy policy instead of determining which slot is optimal. In each time step almost any schedule is possible, because you do not know how many arrivals of type 0, 1 and 2 will occur. So for each time step and for each possible schedule the value function must be computed. At the last time step, where no events can happen anymore, the value function must be zero for each schedule J:

$$V_M(J) = 0. \tag{2.8}$$

For other time steps the value function is as follows:

$$V_j(J) = p_0 * V_j(J|type\ 0) + p_1 * \sum_{k=1}^{N} \left(p_k^1 * V_j(J|k, type\ 1)\right) + p_2 * \sum_{k=1}^{N-1} \left(p_k^2 * V_j(J|k, type\ 2)\right). \tag{2.9}$$

The function consists of three parts, a part for each type of event. As we mentioned before, p_0, p_1 and p_2 are the probabilities for the three different events. $V_j(J|type\ 0)$ are the costs at time j ($j = 0,1, \dots, M$), given that the current schedule is J and the event is of type 0. For type 1 and type 2 events, the values depend on the preference k of the patient. Thus there is a summation for $k = 1, .. N$ for type 1 and for $k = 1, \dots, N-1$ for type 2. Namely, for type 2 there are $N-1$ combinations of two consecutive slots. Because for an event of type 0 nothing happens, the following holds:

$$V_j(J|type\ 0) = V_{j+1}(J). \tag{2.10}$$

For type 1 you have to take the direct costs of assigning slot l according to the greedy policy and the expected value from the next time step on of schedule J without slot l:

$$V_j(J|k, type\ 1) = f(l,k) + V_{j+1}(J\backslash\{l\}). \tag{2.11}$$

For type 2 it is the same as type 1, but now you have to take the expected value of schedule J without slot l and slot $l+1$:

$$V_j(J|k, type\ 2) = f(l,k) + V_{j+1}(J\backslash\{l, l+1\}). \tag{2.12}$$

If the preferred slot k is available, the costs of assigning will be equal to zero and you only add the expected value from the next time step on. If there is no slot available for a patient of type 1, the patient is refused with costs R_1. Once there are no two consecutive slots available for a patient of type 2, the patient is also refused but now with costs R_2. Once a patient is refused, the assigning costs $f(l,k)$ are replaced with the refusal costs. By using these formulas we can calculate $V_0(\emptyset)$, where $\emptyset$ is the schedule where each slot is empty. The pseudo code of this process is shown on the next page.

Algorithm 1 Greedy model

Input: number of slots N, number of time steps M, probabilities for all events: p_0, p_1 and p_2, costs of refusing type 1 and type 2: $R1$ and $R2$, cost function $f(l,k)$ and preference function $g(x)$.

Output: value $V_0(\emptyset)$.

1: Create array with all possible schedules
2: **for** each schedule J **do**
3: **Save:** $V_M(J) = 0$
4: **end for**
5: **for** each time step $j = M-1, \dots, 0$ **do**
6: **for** each schedule J **do**
7: **Save:** $V_j(J|type\ 0) = V_{j+1}(J)$
8: **for** each preference $k = 1, \dots, N$ **do**
9: **if** (no slot available) **do**
10: **Compute** $p_k^1 * V_j(J|k, type\ 1) = p_k^1 * \left(R_1 + V_{j+1}(J\backslash\{l\})\right)$
11: **else**
12: Determine which slot l to assign for type 1
13: **Compute** $p_k^1 * V_j(J|k, type\ 1) = p_k^1 * \left(f(l,k) + V_{j+1}(J\backslash\{l\})\right)$
14: **end if**
15: **end for**
16: **for** each preference $k = 1, \dots, N-1$ **do**
17: **if** (no two consecutive slots available) **do**
18: **Compute** $p_k^2 * V_j(J|k, type\ 2) = p_k^2 * \left(R_2 + V_{j+1}(J\backslash\{l, l+1\})\right)$
19: **else**
20: Determine which slots $l, l+1$ to assign for type 2
21: **Compute** $p_k^2 * V_j(J|k, type\ 2) = p_k^2 * \left(f(l,k) + V_{j+1}(J\backslash\{l, l+1\})\right)$
22: **end if**
23: **end for**
24: **Save:** $V_j(J) = p_0 * V_j(J|type\ 0) + p_1 * \sum_{k=1}^{N}\left(p_k^1 * V_j(J|k, type\ 1)\right) + p_2 * \sum_{k=1}^{N-1}\left(p_k^2 * V_j(J|k, type\ 2)\right)$
25: **end for**
26: **end for**
17: **Return:** expected value $V_j(J)$ for each time step j and schedule J.

2.3 Dynamic programming model

With the dynamic programming (DP) model we want to find the optimal policy. The model works similar as the greedy model, but now the decision of which slot to assign depends on which action will give the lowest expected value. So for each time step, for each schedule J, for each type of patient and each preference the value is computed for each action. This way the model computes which action gives the lowest value and then is the optimal action. Because the model works similar as the greedy model, Formulas (2.8), (2.9) and (2.10) are the same. However, for type 1 and type 2, the following formulas replace Formulas (2.11) and (2.12) from the greedy model:

$$V_j(J|k, type\ 1) = \min\left\{\min_{l\in J}\{f(l,k) + V_{j+1}(J\backslash\{l\})\}, R_1 + V_{j+1}(J)\right\}. \quad (2.13)$$

Schedule $J\backslash\{l\}$ is the same as schedule J, but where slot l is occupied. The same as with $J\{l, l+1\}$ where slot l and slot $l+1$ are occupied.

$$V_j(J|k, type\ 2) = \min\left\{\min_{\{l,l+1\}\in J}\{f(l,k) + V_{j+1}(J\backslash\{l, l+1\})\}, R_2 + V_{j+1}(J)\right\}. \quad (2.14)$$

You can see here that you first have to determine which slot gives the minimum expected value, and also take the minimum of that value and the costs of a potential refusal. This way, it may be better to refuse a patient than to schedule it with high costs. The pseudo code is now as shown on the next page.

Algorithm 2 Dynamic programming model

Input: number of slots N, number of time steps M, probabilities for all events: p_0, p_1 and p_2, costs of refusing type 1 and type 2: $R1$ and $R2$, cost function $f(l,k)$ and preference function $g(x)$.

Output: value $V_0(\emptyset)$ and the optimal slot to assign at time step j with schedule J and an arrival of type 1 or 2.

1: Create array with all possible schedules
2: **for** each schedule J **do**
3: **Save:** $V_M(J) = 0$
4: **end for**
5: **for** each time step $j = M-1, \dots, 0$ **do**
6: **for** each schedule J **do**
7: **Save:** $V_j(J|type\ 0) = V_{j+1}(J)$
8: **for** each preference $k = 1, \dots, N$ **do**
9: **for** each free slot l in J **do**
10: **Compute** $f(l,k) + V_{j+1}(J \backslash \{l\})$
11: **end for**
12: **Compute** $p_k^1 * V_j(J|k, type\ 1) =$
$p_k^1 * \min\{min_{l \in J}\{f(l,k) + V_{j+1}(J \backslash \{l\})\}, R_1 + V_{j+1}(J)\}$
13: **Save:** the optimal slot l to assign
14: **end for**
15: **for** each preference $k = 1, \dots, N-1$ **do**
16: **for** all free slots $l, l+1$ in J **do**
17: **Compute** $f(l,k) + V_{j+1}(J \backslash \{l, l+1\})$
18: **end for**
19: **Compute** $p_k^2 * V_j(J|k, type\ 2) =$
$p_k^2 * \min\{min_{\{l,l+1\} \in J}\{f(l,k) + V_{j+1}(J \backslash \{l, l+1\})\}, R_2 + V_{j+1}(J)\}$
20: **Save:** the optimal slots $l, l+1$ to assign
21: **end for**
22: **Save:** $V_j(J) = p_0 * V_j(J|type\ 0) + p_1 * \sum_{k=1}^{N} \left(p_k^1 * V_j(J|k, type\ 1)\right) + p_2 * \sum_{k=1}^{N-1} \left(p_k^2 * V_j(J|k, type\ 2)\right)$
23: **end for**
24: **end for**
25: **Return:** expected value $V_j(J)$ for each time step j and schedule J and the optimal slot to assign at time step j with schedule J and an arrival of type 1 or 2.

2.4 Reduced dynamic programming model

The DP model calculates the policy for each time step, each schedule and each preference. With the DP model there are 2^N possible schedules. As the number of slots grows, the number of possible schedules grows tremendously. This caused a huge increase of calculation time for the model. If the schedule consists of 13 slots, the calculation time is a few hours. Therefore, the reduced DP model has less states.

The idea of this model is to split the schedule in clusters instead of slots. First you have to remember for each slot in a schedule if it was occupied or not. If the schedule consists of 20 slots this method gave $2^{20} = 1.048.576$ different possible schedules. If you split the schedule in five clusters of four slots you only have to remember how many slots in a cluster are taken. There are now five states for each cluster: zero, one, two, three or four occupied slots. This will give $5^5 = 3.125$ different schedules. This is a huge difference and will solve the exponentially increasing calculation time. But if you do not know which slots are exactly available, how do you know if it is possible to plan a patient in that cluster and what are the resulting costs?

Each patient now arrives with a preference for a specific cluster. If the patient is assigned to another cluster than the one he/she prefers, costs are made. The costs are the computed with the formula $f_1(l, k)$, where l is the assigned cluster and k the preferred cluster. As a patient is assigned to a cluster again there are costs involved. The costs depend on the number of occupied slots in the cluster and the preferred slot of the patient, which you do not know anymore. These type of costs are described as $C(i, l)$, where i is the type of patient and l the assigned cluster.

To estimate the costs $C(i, l)$ we prescribed a strategy. The strategy for each cluster is described in Appendix A. For example, if the cluster is free and a patient of type 1 arrives, you give the slot he/she prefers. Therefore, the costs will be zero. But if a patient of type 2 arrives you place him at the left of the cluster or at the right (see two schedules below), but not in the middle. This way, there are two slots free next to each other, for a type 2 patient that may arrive in the future. If the patient has a preference for slots 1 + 2 or 3 + 4, there will be no costs. But if the patient has a preference for slot 2 + 3, there will be costs. So in this case you take the costs of assigning a patient with a preference for slot 2 + 3 and divide it by three. We can only divide it by three if we use the uniform distributed preference function. This will be the estimated costs for placing a patient of type 2 in a cluster with 4 empty slots. Because these costs are an estimation, the actual costs may differ. Therefore, it will be difficult to compare the results of this model with the Greedy and DP model. Also, it is not possible anymore to place a patient of type 2 at the slot 4 and slot 1 of two adjacent clusters. This can affect the resulting values.

1	2	3	4
■	■		

1	2	3	4
		■	■

The pseudo code of this model is similar the pseudo code of the DP model. The difference is that a schedule now consists of clusters instead of slots. The state of each cluster is part of $\{0,1,2,3,4\}$ instead of $\{0,1\}$ for each slot. Also, the model now only works with the linear preference function, because you have to add up two components in the costs. As a patient is assigned to a cluster he does not prefer, costs $f_1(l, k)$ are made. Next there are costs $C(i, l)$ made by placing the patient in the cluster. Therefore, only the linear cost function can be used. The current implementation only works with clusters of size four, so the number of slots has to be a multiple of four. In the pseudo code you will find the variables $J\{l^*\}$ and $J\{l^{**}\}$. These variables mean that the cluster l in schedule J has been assigned an appointment of patient type 1 or type 2.

Algorithm 3 Reduced dynamic programming model

Input: number of slots N, number of time steps M, probabilities for all events: p_0, p_1 and p_2, costs of refusing type 1 and type 2: $R1$ and $R2$, expected costs for type i and cluster l: $C(i,l)$.

Output: value $V_0(\emptyset)$ and the optimal cluster to assign at time step j with schedule J and an arrival of type 1 or 2.

1: Create array with all possible schedules
2: **for** each schedule J **do**
3: **Save:** $V_M(J) = 0$
4: **end for**
5: **for** each time step $j = M-1, \dots, 0$ **do**
6: **for** each schedule J **do**
7: **Save:** $V_j(J|type\ 0) = V_{j+1}(J)$
8: **for** each preference $k = 1, \dots, \frac{N}{4}$ **do**
9: **for** each available cluster l in J **do**
10: **Compute** $f_1(l,k) + C(1,l) + V_{j+1}(J\{l^*\})$
11: **end for**
12: **Compute** $p_k^1 * V_j(J|k, type\ 1) = p_k^1 * \min\{min_{l \in J}\{f_1(l,k) + C(1,l) + V_{j+1}(J\{l^*\})\}, R_1 + V_{j+1}(J)\}$
13: **Save:** the optimal cluster l to assign
14: **end for**
15: **for** each preference $k = 1, \dots, \frac{N}{4}$ **do**
16: **for** each available cluster l in J **do**
17: **Compute** $f_1(l,k) + C(2,l) + V_{j+1}(J\{l^{**}\})$
18: **end for**
19: **Compute** $p_k^2 * V_j(J|k, type\ 2) = p_k^2 * \min\{min_{\{l\} \in J}\{f_1(l,k) + C(2,l) + V_{j+1}(J\{l^{**}\})\}, R_2 + V_{j+1}(J)\}$
20: **Save:** the optimal cluster l to assign
21: **end for**
22: **Save:** $V_j(J) = p_0 * V_j(J|type\ 0) + p_1 * \sum_{k=1}^{\frac{N}{4}} \left(p_k^1 * V_j(J|k, type\ 1)\right) + p_2 * \sum_{k=1}^{\frac{N}{4}} \left(p_k^2 * V_j(J|k, type\ 2)\right)$
23: **end for**
24: **end for**
25: **Return:** expected value $V_j(J)$ for each time step j and schedule J and the optimal cluster to assign at time step j with schedule J and an arrival of type 1 or 2.

3. Results

This chapter describes the validation and the results of the models. We describe how we validated the greedy and DP model using Ottens' report in section 3.1. Also, we present the results obtained from the greedy and the DP model in section 3.3 and the results of the reduced DP model in section 3.4.

3.1 Validation

First step of this research was to validate the results with the ones Ottens retrieved in her research. Ottens' model only had one type of patient, no refusals and each time step a patient wanted to make an appointment. So at the end the schedule will be completely full. For to this model, we had to choose the probabilities as follows: $p_0 = p_2 = 0$ and $p_1 = 1$. Also, the number of time steps M is now equal to the number of slots N. This is only necessary for the greedy model so that each time step a patient of type 1 arrives. For the other models M is determined such that the schedule is expected to be 95% full. The refuse costs were set extremely high, so it will never be better to refuse a patient than assigning a slot in the schedule.

Ottens calculated the optimal value of the greedy and DP model for each preference and cost function and for $N = 5$. This resulted in two tables. It was checked that our program gave the same values. The tables can be found below.

	$f_1(l,k)$	$f_2(l,k)$	$f_3(l,k)$
$g_1(x)$	0.6596	0.2847	0.5848
$g_2(x)$	0.7483	0.3210	0.6729
$g_3(x)$	0.8417	0.2995	0.7085

Table 1: Optimal values greedy model, only type 1

	$f_1(l,k)$	$f_2(l,k)$	$f_3(l,k)$
$g_1(x)$	0.6513	0.2644	0.5698
$g_2(x)$	0.7427	0.3099	0.6629
$g_3(x)$	0.8050	0.2282	0.5857

Table 2: Optimal values DP model, only type 1

Besides the optimal values, Ottens also presented several examples of the appointment scheduling process. It was checked if our program showed the same behavior as these examples and most of the time it did. For example, we use $g_1(x)$, the first function for the preference distribution, and the quadratic cost function $f_2(l,k)$. The current schedule is as follows:

1	2	3	4	5

If a patient now arrives with a preference for slot 3, slot 2 will be given to retain the dispersion of the free slots. In this situation, our program gave the same results as Ottens. In some other examples, our program did not gave the same results. Now we use the second function for the preference $g_2(x)$ and the linear cost function $f_1(l,k)$. The current schedule is as follows:

1	2	3	4	5

If a patient arrives with a preference for slot 2, according to Ottens slot 3 should be assigned. According to our program, slot 1 is assigned. Because we use $g_3(x)$, it is more likely that the next patients will have a preference for the middle slots. Therefore it makes more sense to keep the middle slot free. It is difficult to find out where the differences in these two examples come from. But since most examples gave the same results and the optimal values were exactly the same, we assume that the greedy and DP model give correct results.

3.2 Results dynamic programming model

First, we compare the optimal values of the greedy and the DP model. In Ottens' report, the differences between the two models were fairly small, but the DP model was always better than the greedy model. Since the model in Ottens' report is not stochastic and only uses a patient of type 1, we expect that the differences are now larger. Table 3 shows the differences between the optimal values of the two models for the different preference and cost functions and $N = 5$. The probabilities for each type of event are as follows: $p_0 = 0.2$, $p_1 = 0.5$ and $p_2 = 0.3$. The refusal costs are: $R_1 = 1$ and $R_2 = 2$. As you can see, the values differ per combination of functions and the DP model is always better than the greedy model. Table 4 shows the same differences, but now for a schedule with $N = 10$. The probabilities for the events and the refusal costs are the same. As expected, the differences are higher for a schedule with more slots. This means, that as the size of the schedule increases, the absolute difference between the optimal and greedy policy becomes larger.

	$f_1(l,k)$	$f_2(l,k)$	$f_3(l,k)$
$g_1(x)$	-0.0544	-0.1434	-0.0966
$g_2(x)$	-0.0267	-0.0861	-0.0518
$g_3(x)$	-0.1559	-0.2467	-0.2152

Table 3: Difference between DP and greedy model with N = 5 and M = 4

	$f_1(l,k)$	$f_2(l,k)$	$f_3(l,k)$
$g_1(x)$	-0.1441	-0.2477	-0.2142
$g_2(x)$	-0.0754	-0.1265	-0.1192
$g_3(x)$	-0.1924	-0.2849	-0.2698

Table 4: Difference between DP and greedy model with N = 10 and M = 9

Figure 2 shows the optimal value of different models for different schedule sizes. Here, the cost function is $f_1(l,k)$ and the preference function is $g_1(x)$. The probabilities for the different types of events and the refusal costs are unchanged. Unfortunately, the calculation time became too big for N larger than twelve, so we decided not to calculate the values anymore. As stated in Chapter 2, the value function gives an indication of how high the costs will be if you use this model. As you can see in the graph, the values of the DP model are always lower than the greedy model and therefore better. Also, it appears that for a bigger schedule, the difference between the two values also gets bigger. The graph has a lower point at $N = 12$. The number of time steps M is always chosen such that the schedule will be at least occupied for 95%. If we use the same probabilities p_0, p_1 and p_2 then up until $N = 11$ this means

that $M = N - 1$. But when N reaches the size of twelve, M should be equal to ten. This causes a lower for the value.

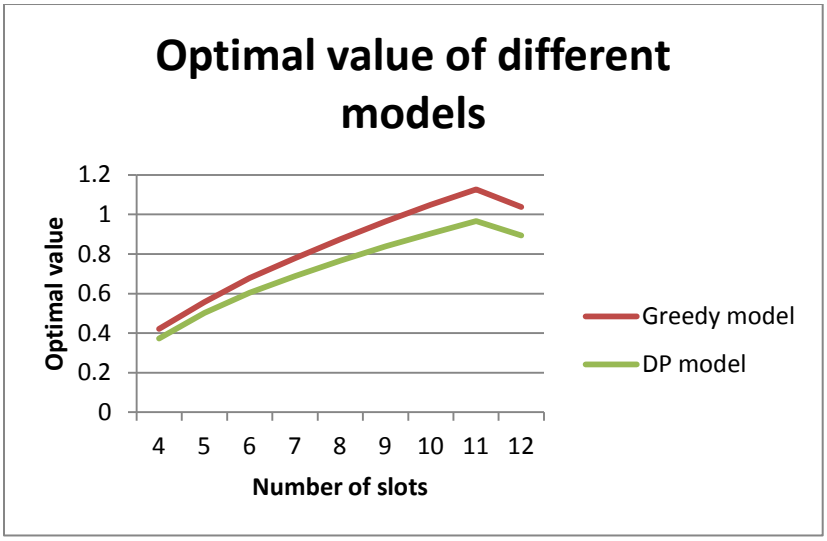


Figure 3: Optimal value of different models

To show the behavior of the DP model a few examples will be presented for a schedule with five slots. The examples are all for a linear cost function and a uniform distribution for the preference function.

Example 1: In this example we start with the schedule that is shown below.

1	2	3	4	5

The table below shows which slot will be assigned to a patient of type 1, for preferences 1,…,5 presented at the top row and for $M = 4, \ldots, 1$. This means that at $M = 4$, there are still 4 time steps to assign patients. Thus $M = 1$ is the last time to assign patients.

	1	**2**	**3**	**4**	**5**
M = 4	1	2	4	4	4
M = 3	1	4	4	4	4
M = 2	1	4	4	4	4
M = 1	1	2	2	4	4

Table 5: Example 1, patient type 1

The table shows that the first time step, the patient gets the slot that he prefers or closest to the one he/she prefers. If a patient arrives with a preference for slot 3, slot 4 is assigned. You might except that slot 2 is assigned, because the costs of placing at slot 2 and slot 4 are the same and the model then has a preference for the left slot. But now apparently it is better to keep the slots 1 and 2 free for a patient of type 2 that may arrive in the future. At $M = 3$, this behavior starts to change. If a patient arrives with a preference for slot 2, slot 4 is assigned. It seems as if the model again is trying to keep slots 1 and 2 free, for patients of type 2, that may arrive in the future. It is remarkable that this behavior did not show at

the first time step. At the last time step, the model just gives the patient what he wants, because there aren't any time steps to come. For type 2, the model does not show a change in behavior over time. The patient is always assigned to slot 1 and 2.

Example 2: In this example we start with the following schedule:

1	2	3	4	5

The table below again shows which slot will be assigned to a patient of type 1, for preferences 1,…,5 and for $M = 4, \dots, 1$.

	1	**2**	**3**	**4**	**5**
M = 4	2	2	3	4	5
M = 3	2	2	3	4	5
M = 2	2	2	3	4	5
M = 1	2	2	3	4	5

Table 6: Example 2, patient type 1

In this example the behavior of scheduling a patient of type 1 does not change over time. Perhaps the reason is that after scheduling a patient of type 1, you will always have enough space of planning another patient of type 1 and of type 2. The behavior of scheduling a patient of type 2 does change over time as you can see in table 7. In the first three time steps, the patient is always placed at slot 2 + 3 or at slot 4 + 5. This way, there is still room for another patient of type 2. Of course at the last time step, you always give the patient the (nearest) slots he prefers.

	1	**2**	**3**	**4**
M = 4	2	2	4	4
M = 3	2	2	4	4
M = 2	2	2	4	4
M = 1	2	3	4	4

Table 7: Example 2, patient type 2

Figure 2 shows us that the difference between the values of the greedy and DP model grows as the number of slots increases. We have created six different scenarios to investigate what other factors may have an influence on this difference. Table 8 shows these five scenarios. Scenario 1 is the same as we used for figure 2. In scenario 2 the probability of an arrival of patient type 2 is higher. Therefore, the number of time steps M is lower, because it will take less time steps to fill up the schedule. In scenario 3 and scenario 4 the refusal costs are higher. In scenario 5 the probability of an arrival of patient type 1 is higher. Finally, in scenario the probability of an arrival of patient type 2 is much higher. Here, the number of time steps M is also much higher.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
p_0	0.2	0.1	0.1	0.1	0.1	0.9
p_1	0.5	0.3	0.3	0.3	0.8	0.05
p_2	0.3	0.6	0.6	0.6	0.1	0.05
N	10	10	10	10	10	10
M	9	6	6	6	10	63
R_1	1	1	2	5	5	5
R_2	2	2	4	10	10	10

Table 8: Different scenarios

Figure 2 shows the results of the five scenarios. In all scenarios the DP model gives better results than the greedy model. As you can see, the difference becomes larger when there are more patients of type 2 and when the refusal costs are higher. This means that if you have more patients of type 2 and/or you have high refusal costs, the gain you get from using the DP model will increase.

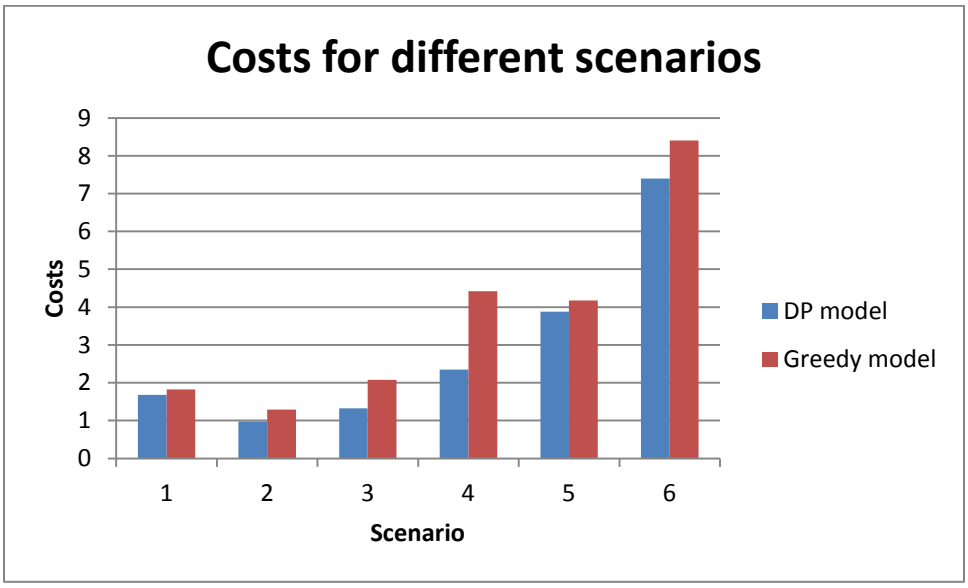


Figure 2: costs per scenario

3.3 Results reduced dynamic programming model

An important part of the reduced DP model was that it should have a small calculation time and is able to determine the policies for a schedule up to size twenty. Table 9 below shows the calculation time in minutes of the three models for N divisible by four. Unfortunately, performing the greedy and the DP model takes too much time for $N \geq 16$. But the table shows that the calculation time of the reduced DP model is very small and it can determine the policies of a schedule of size twenty in a minute. In this section, each result comes from performing the models with the uniform preference function $g_1(x)$ and linear cost function $f_1(l, k)$.

N	M	Greedy model	DP model	Reduced DP model
4	**3**	0.0006	0.0007	0.0005
8	**7**	0.0074	0.0112	0.0012
12	**10**	1.2930	6.5773	0.0047
16	**14**	> 4 hrs	> 4 hrs	0.0236
20	**17**	> 4 hrs	> 4 hrs	1.0160

Table 9: Calculation time in minutes

To give an idea of how the reduced DP model works an example is presented. The example considers a schedule with twenty slots divided into five clusters of four slots and use the same event probabilities and refusal costs as in scenario 1.

Example 3: The schedule now looks as follows:

1	2	3	4	5
0	1	2	3	4

The schedule shows the five clusters with all different numbers of taken slots. Now, we do not look at the slot the patient prefers, but the cluster. The table below shows which cluster will be assigned to a patient of type 1, for preference 1,…,5 and for $M = 17, \dots, 1$.

	1	2	3	4	5
M = 17	1	2	R	4	R
M = 16	1	2	R	4	R
M = 15	1	2	R	4	R
…					
M = 3	1	2	3	4	4
M = 2	1	1	3	4	4
M = 1	1	1	3	4	4

Table 10: Example 3, patient type 1

This example shows some remarkable actions. The first time steps for a patient of type 1 with a preference for cluster 3 or cluster 5, he/she will be refused. For a preference for cluster 5 this is logical because you cannot be assigned to cluster 5 so you already have the costs of placing the patient in a cluster he does not prefer. For the patient with a preference for cluster 3 the first time steps he/she is probably is refused so this cluster is still available for a patient of type two. Also, it probably is cheaper to refuse this patient then to place he/she at cluster 2 or cluster 4. This behavior changes at a certain point.

At the last time steps also something remarkable happens. At $M = 3$ a greedy policy occurs, which is logical at the last time steps. But this is different for $M = 2$ and $M = 1$. Then a patient with a preference for cluster 2 is assigned to cluster 1. This may have to do with the fact that placing a patient in a cluster with one occupied slots costs more than placing it in an empty cluster.

Table 11 shows the policy for a patient type 2. At the first time steps a patient with a preference for cluster 5 is refused. At the last time steps this behavior is changed. The patient with a preference for

cluster 5 is now placed at cluster 3. It seems as if the model tries to does not fill cluster 3 completely at first, but later on it does not matter anymore because the day is almost over.

	1	2	3	4	5
M = 17	1	2	3	3	R
M = 16	1	2	3	3	R
M = 15	1	2	3	3	R
…					
M = 3	1	2	3	3	3
M = 2	1	2	3	3	3
M = 1	1	2	3	3	3

Table 11: Example 3, patient type 2

Table 12 below shows the values of the three models for values of N that are divisible by four. Unfortunately, we cannot compute the values for the greedy and DP model for $N = 16$ and $N = 20$, because of the large computation time. But it is striking that the reduced DP model gives higher values than the greedy and the DP model. This means that the reduced DP model seems to perform worse than the other two models. The optimal value of the reduced DP model is an estimation because we do not know which slot the patient really prefers and gets assigned. We expect that the true values of the reduced DP model are lower, because you add up costs of placing in another cluster and costs of placing inside the cluster. But we do not know how much lower they will be. Further research is necessary to make strong conclusions about the performance of the reduced DP model.

N	M	Greedy model	DP model	Reduced DP model
8	**7**	1.4589	1.3499	1.878
12	**10**	1.4966	1.352	2.3344
16	**14**			3.3922
20	**17**			3.7131

Table 12: Optimal values of reduced DP model

4. Conclusion and recommendations

The purpose of this research paper was to investigate what the best way was of planning patients with a preference in an outpatient department. Three models were implemented: the greedy model, the dynamic programming and the reduced dynamic programming model. The results show that the DP model always gives a lower value and thus lower costs than the greedy model. Therefore we can conclude that the DP model is actually better than the greedy model.

Since a patient of type 2 needs two slots, logically you want to keep two slots free next to each other. The model indeed shows this behavior. Since the patient of type 2 is a new patient, which is really important to a hospital, this is probably the point where the DP model is better than the greedy model. As there are more patients of type 2 and the refusal costs become higher, the DP model becomes even more better than the greedy model. Besides, sometimes it is advantageous to refuse a patient at the beginning of the planning process. Apparently it is beneficial to keep these slots free for future patients. Adding another patient type and the option to refuse patients, clearly influences the planning process.

Because of the exponentially increasing calculation time of the DP model, the reduced DP model was developed. The reduced DP model does not remember for each slot if it is occupied or not, but it splits the schedule up in clusters. Next, for each cluster is remembered how many slots are occupied in this cluster. This way, the calculation time is greatly reduced. Unfortunately the values returned by the reduced DP model are higher than those of the greedy and the DP model. The values of the reduced DP model are an estimation. They do not represent the actual costs. Because the reduced DP model does not know which slots are exactly occupied it is difficult to compute the actual costs. An expansion of the program is necessary, which we have not achieved because of lack of time. Therefore it is recommended to expand the program in future research. For a schedule of eight or twelve slots, divided in clusters of size four, this must be possible to compute. We expect that the actual costs may be lower than they are now, because there are two types of costs added up: the costs of placing a patient in a cluster that he does not prefer plus the expected costs of placing the patient in the cluster. Future research will have to show how low the actual costs are.

Based on the results of this research, we would advise the outpatient department to do more research for the reduced DP model. The results show that the profit you can get by using the DP model could be around 47%. We do not know what the exacts costs of the reduced DP model are, but after further research it can be decided to implement the reduced DP model if it turns out to be more profitable. This really depends on the refusal costs and the distribution between patients of type 1 and of type 2. In addition, it may be useful to take with you what the costs are of empty slots at the end of the day. The number of time steps are now chosen such that the schedule is at least 95% full at the end of the day. Moreover, the program can be extended for multiple days. Then the patient will not be refused, but shifted to the next day. If all this is taken into account in the improved model, it may turn out to be that the reduced DP model is better than the greedy model.

Bibliography

[1] L. Ottens, Appointment Scheduling with Patient Preference, Research Paper, VU University Amsterdam, August 4, 2013.

Appendix A: strategy per cluster

Number of taken slots = 0:

1	2	3	4

- Type 1: give the slot he/she prefers
- Type 2: give slot 1 + 2 or at slot 3 + 4, depending on the preference

Number of taken slots = 1:

1	2	3	4
■			

- Type 1: give slot 2 or slot 4, depending on the preference
- Type 2: give the two slots closest to the preference

1	2	3	4
	■		

- Type 1: give slot 1
- Type 2: give slot 3 + 4

1	2	3	4
		■	

- Type 1: give slot 4
- Type 2: give slot 1 + 2

1	2	3	4
			■

- Type 1: give slot 1 or slot 3, depending on the preference
- Type 2: give the two slots closest to the preference

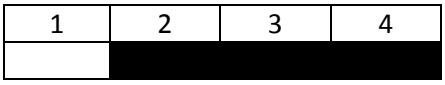
Number of taken slots = 2:

1	2	3	4
■	■		

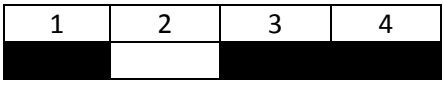
- Type 1: give the slot closest to the preference
- Type 2: give slot 3 + 4

- Type 1: give the slot closest to the preference
- Type 2: give slot 1 + 2

Number of taken slots = 3:

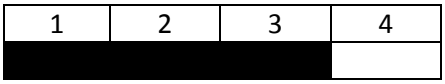


- Type 1: give slot 1
- Type 2: refuse



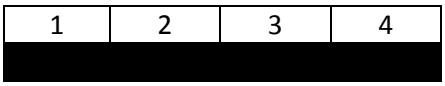
- Type 1: give slot 2
- Type 2: refuse

- Type 1: give slot 3
- Type 2: refuse



- Type 1: give slot 4
- Type 2: refuse

Number of taken slots = 4:



- Type 1: refuse
- Type 2: refuse

Appendix B: Binary representation

Index	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5
0					
1					X
2				X	
3				X	X
4			X		
5			X		X
6			X	X	
7			X	X	X
8		X			
9		X			X
10		X		X	
11		X		X	X
12		X	X		
13		X	X		X
14		X	X	X	
15		X	X	X	X
16	X				
17	X				X
18	X			X	
19	X			X	X
20	X		X		
21	X		X		X
22	X		X	X	
23	X		X	X	X
24	X	X			
25	X	X			X
26	X	X		X	
27	X	X		X	X
28	X	X	X		
29	X	X	X		X
30	X	X	X	X	
31	X	X	X	X	X

An ‘X’ implies that the slot is occupied. As you can see there are $2^5 = 32$ different possible schedules.