BSc Thesis Applied Mathematics

Patient scheduling optimization through an application of the cutting stock problem

Peter Klein Kranenbarg

Supervisor: dr.ir. Maartje van de Vrugt

January, 2020

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

This paper was written to fulfill the graduation requirements of the Bachelor Applied Mathematics at the University of Twente. The research was performed from September 2, 2019 up to January 24, 2020.
The research was designed to help out the neurology and neurosurgery wards at the VUmc. They provided the data needed for the case study.
I would like to thank my supervisor for giving critical feedback, helping with the research, setting up the contacts with the VUmc and her general guidance during the research.
Moreover, I would like to thank my friends, family, roommate and fellow students for their mental support.
Anna, Arnout, Bram, Jasper, who I have been working with in the Educafe the past semester I would like to thank for providing a good work environment, and for occasionally helping me out with my research.
Also, I would like to thank my father for critically reading my thesis, helping me out to (hopefully) remove the final mistakes.
I very much enjoyed performing the research, and I hope you enjoy your reading.

Peter,
January 24, 2020

Contents

Patient scheduling optimization through an application of the cutting stock problem

Peter Klein Kranenbarg

January, 2020

Abstract

Efficiently using hospital beds is increasingly important, as healthcare expenditures are constantly rising. In this paper we present a research on this topic, taking into account both unscheduled and scheduled admissions. We used a modified version of the cutting stock problem to define an integer linear program model and tested our results in two simulations. In a case study on the VUmc neurology and neurosurgery department we achieved strategies requiring 28 beds for a blocking probability of 5% which is a significant improvement on the current 30 beds required by the VUmc for the same blocking probability.
Keywords Cutting stock problem · Discrete event simulation · Integer Linear Program · Optimization · Health care · Resource allocation · Refused admissions · Bed pooling

1 Introduction

The problem of efficiently planning patients at a hospital has been researched for many years. It is often referred to as the Patient Admission Scheduling Problem (PASP) and is more formally stated as the problem of assigning patients to beds over a given time horizon so as to maximize efficiency and patient comfort while satisfying the necessary (medical) constraints. The name PASP was introduced in 1952 by T.J. Bailey [1], who addressed the problem as a queuing system with the objective of minimizing patients' waiting time. After this the PASP has been researched extensively. Granja, C. et al [8] used simulated annealing to tackle an admission scheduling problem in a private imaging clinic where all admissions are scheduled. Bastos et al. [2] used mixed integer programming and considered only scheduled admissions, too. Other approaches that have succeeded are: stochastic integer programming [11], a hybrid of the variable neighbourhood descent method and the iterated local search method [9], mixed integer programming based heuristics [17] and a non-linear great deluge algorithm [10], amongst other methods. Often these methods were invented for a specific case, most of the time resulting in models that only consider scheduled admissions. In reality, however, many hospital wards have to deal with unscheduled admissions as well.

The research that we present in this paper was designed for a situation at the VUmc in Amsterdam, where the head of the neurology and neurosurgery wards wants to know if it is possible to merge, since they have been working together for a long time. Because more than half of the patients at these wards arrive unscheduled, most existing models are not suited to investigate this possibility.

The main problem for the two wards to merge is the lack of capacity at each ward individually. Together they now require at least 30 beds to treat all patients, but the physical capacity of the wards is 28 beds. Therefore our main research goal is to find the minimum capacity one ward would need for the two wards to be able to merge. In order to find this capacity we had to take three steps:

1. Data analysis.
2. Optimizing the patient scheduling policy.
3. Testing our results in a simulation.

Which resulted in the following research goals:

1. (a) Determine the best fitting probability distribution for the patients characteristics.
   (b) Use a queuing model to determine the current blocking probabilities for given capacities at the hospital.
2. Optimize the scheduling strategy using an Integer Linear Programming Model.
3. Investigate through a simulation the practical relevance and robustness of the strategies.

1.1 Literature

1.1.1 Data analysis

After finding fitting distributions to the characteristics of the patients, we investigated the current situation at the VUmc through a queuing model. Queuing models are not used as much in health care as in industry, but there are some interesting articles on necessary capacities and blocking probabilities in hospitals using queuing models, one of which we briefly describe below:

Dimensioning hospital wards using the Erlang loss model [4]

This article is an important source of inspiration for our research, since the application is similar to our situation. They investigate the situation at a hospital with different wards which can possibly merge. After investigating the data, they conclude that a Poisson distribution is fitting for the combination of emergency and elective admissions, and with that, that an Erlang loss model describes the situation accurately. They use this model to investigate occupancy rates and blocking probabilities for certain capacities. For this type of model only the mean of the length of stay is relevant, which makes it easy to implement. In their paper, they cite an article from Young (1965) [22], which shows that the assumption of a Poisson arrival is realistic for unscheduled admissions. This paper dates back quite a few years, but it is reasonable to assume that, although the amount of patients and their specific required treatments might have changed, the pattern of unscheduled admissions has not changed over the last 50 years. Therefore this paper could be useful to argue assumptions in our model.

Phase-type distribution

The assumption of Poisson arrivals for elective admission is not generally accurate, for example because elective admissions occur more during the day than during the night. A Phase-type distribution is constructed by a convolution of exponential distributions. It

results from a system of one or more Poisson processes occurring in sequence. The sequence in which the different Poisson processes occur may itself be a stochastic process [19]. Since the variability in elective admissions can be captured well by the Poisson distribution [4] and the Phase-type distribution could take into account the change in the arrival rate over time (mainly the difference between weekdays and the weekend), the Phase-type distribution might well describe the arrival process of elective admissions accurately. This has been introduced to the context of hospital bed allocation by F. Gorunescu et al. [6] [7] in 2002. Later, in 2009, X. Li et al. [12] used a similar queuing model as basis for a multi-objective decision aiding model to improve bed allocation in a hospital in China.
To our knowledge, there are no analytical results for $Ph/Ph/c$ queues, which based on the data analysis of [4] could be the best fitting queuing model. There has however been research on algorithms for numerical analysis of multi-server queuing models by Takahashi [15] and Takahashi and Takami [16]. Later, there has been other research on the same subject, for instance by L.P. Seelen [14].
Another queuing system that might fit best is the $Ph/G/c$ queuing system. On this queuing system, however, there are no papers to our knowledge. If a general distribution for the length of stay is the only good fit to the data, also a $G/G/c$ queue could be considered. This queuing system also has no analytical results, but has been investigated to find approximations by for example J.S. Wu [21] and D. Bertsimas [3]. The approximations of the $Ph/Ph/c$ and $G/G/c$ queuing systems are quite complicated and computationally heavy, which makes the assumption of a $M/G/c$ queuing system in the article we described earlier by de Bruin et al. [4] reasonable.

1.1.2 Planning strategy optimization

The model we implemented to improve the planning strategy of the patients that can be freely scheduled is a variant of the cutting stock problem. The cutting stock problem is one of the well known applications of the bin packing problem, a famous problem in combinatorial optimization. The cutting stock problem can be defined formally as follows [5]: "We are given m *item types*, each having an integer *weight* w_j and an integer *demand* d_j ($j = 1, \dots, m$), and a sufficiently large number of *bins* of integer capacity c. The objective is to produce d_j copies of each item type j using the minimum number of bins so that the total weight in any bin does not exceed its capacity."
A paper by Van de Vrugt [20] uses the cutting stock problem to determine a planning strategy for patients and inspired us to use that methodology too. However, where they only considered scheduled admissions, for our implementation of the cutting stock problem we had to consider unscheduled admissions too.
In Section 2 we describe the general model we set up. In Section 3 we present our data analysis, and the model input for our case study. And in Section 4 we present the results of our model implementation for the situation at the VUmc.

2 Model

Our bed usage optimization model is a variant of the cutting stock problem, described in Section 1.1.2. In our situation at a hospital one bed is one bin. Let h be the planning horizon in days, then each bed has a capacity of $24 * h$ "bed hours". Suppose we have m patients, each requiring q_j ($j = 1, \dots, m$) of these "bed hours". Then our objective is to minimize the number of beds required to meet the demand of each patient, i.e. treat each patient.

Let n_t denote the number of different patient types $t = 0, 1, \ldots, n_t - 1$, all patients of one type having the same expected length of stay. Patients that arrive unscheduled are of type 0, which we refer to as the emergency patients. The scheduled admissions we refer to as elective patients. Let l_t denote the required bed hours for a patient of type t, and let the way one bed is utilized over the length of the planning horizon be denoted by a pattern, i.e. an ordered list of patients. Let N_{it} denote the number of patients of type t in some pattern i. Such a pattern is feasible if and only if:

- $\sum_t N_{it} * l_t \leq 24 * h$; the sum of all the utilized bed hours is smaller or equal than the available number of bed hours.
- $\sum_t N_{it} * l_t \geq 24 * h - \min_t\{l_t\}$; there is no room for an extra patient in the pattern.
- There is no elective admission in the weekend.
- There is no elective admission after 6.00PM or before 8.00AM.

We use what we refer to as a "dynamic planning", which means that we allow a pattern to start later than day 0, time 00:00, and leave a patient not fully treated at the end of the planning, as long as that last patient could have been treated in the extra time that is not used at the start. More formally this means the total used "bed hours" of a pattern is the planning period in hours, but this does not necessarily mean that the planning of the pattern starts at day one and ends at the last day of the planning period.

2.1 Creation of the feasible patterns

The patterns were created by starting with one list of patterns, each pattern with one admission planned. Then we used a loop to continuously transform each pattern in n_t new patterns by adding to each pattern a patient of one of the types $0, 1, \ldots, n_t - 1$. We let this process repeat until all patterns in the list are full.

2.1.1 Initialization

All patterns started with one of the following admissions: an emergency admission on one of the first eight days at 00:00, 08:00 or 16:00, or an elective admission on one of the first eight days (excluding the weekend) at 08:00 or 16:00. So this resulted in $8*3+(n_t-1)*6*2$ different patterns to start with.

2.1.2 Pattern creation loop

The pattern creation loop uses one important function that we describe separately: the "insert patient" function. The pseudo code for the "insert patient" function is provided below:

```
def insert_patient(patient, pattern):
    if emergency patient:
        if enough bed hours left:
            add the patient to the pattern
        else:
            pattern is full
    if elective patient:
        if enough bed hours left:
```

```
            if Tuesday-Friday before 8.00 AM:
                add the patient to the pattern at 8.00 AM
            if Monday-Thursday after 6.00 PM:
                add the patient to the pattern next morning at 8.00 AM
            if between Friday 6.00 PM and Monday 8.00 AM:
                add the patient to the pattern at Monday 8.00 AM
            if Monday-Friday between 8.00 AM and 6.00 PM:
                add the patient to the pattern at that time
        else:
            pattern is full
```

Using this function, the pseudo code for the pattern creation loop is as follows:

```
def pattern_creation_loop()
    initialize patterns_list
    while not all patterns full:
        for p in pattern_list:
            make n_t copies of p
            add one patient of type 0 or 1 or ... or n_t-1 to the copies
```

2.2 Integer Linear Program

From the pattern creation loop we obtained a list of feasible patterns $[0, \ldots]$. Let x_i denote the number of times pattern i is used in the solution. These x_i's are then the decision variables of the ILP. The objective is

$$\min \sum_i x_i$$

Let $d_1, \ldots, d_{n_t-1}$ denote the number of patients of type $1, \ldots, n_t - 1$ that need to be treated within the planning period. For the emergency patients, because we can not schedule them, we do not require a number of them to be treated within the planning period, but instead we require a minimum and maximum of available admissions per day, and a minimum of available admissions per week in our schedule. Let e_d denote the minimum number of emergency admissions per day, e_m the maximum number of emergency admissions per day and e_w the minimum number of emergency admissions per week. Moreover, let $c_{i,1}, c_{i,2}, \ldots, c_{i,n_t-1}$ denote the number of patients of type $t = 1, 2, \ldots, n_t - 1$ in pattern i. And let $u_{i,0}, u_{i,1} \ldots, u_{i,h-1}$ denote the number of emergency arrivals planned on day $0, 1, \ldots, h - 1$ in pattern i. Finally, let $v_{i,0}, v_{i,1}, \ldots, v_{i,h/7-1}$ denote the number of emergency arrivals planned in week $0, 1, \ldots, h/7 - 1$ in pattern i. Note that the planning horizon should be divisible by seven. Then the constraints are as follows:

$$\sum_i c_{ij} x_i \geq d_j \quad \text{for} \quad j = 1, \ldots, n_t - 1$$

$$\sum_i u_{ik} x_i \geq e_d \quad \text{for} \quad k = 0, \ldots, h - 1$$

$$\sum_i u_{ik} x_i \leq e_m \quad \text{for} \quad k = 0, \ldots, h - 1$$

$$\sum_i v_{im} x_i \geq e_w \quad \text{for} \quad m = 0, \ldots, h/7 - 1$$

$$x_i \geq 0 \quad \forall i$$

2.3 Simulation

The found strategies from the ILP were tested in a Discrete Event Simulation (DES). The DES consists of a list of events with an execution time attached, which are executed in chronological order. There are four types of events:

- Arrival: A patient arrives at the hospital (scheduled or unscheduled).
- EndService: A patient is discharged from the hospital.
- Measurepoint: A point at which the program notes down statistics.
- EndOfTime: The end of the simulation run.

At the event "EndOfTime" the simulation is stopped. The length of the simulation is 1000 times the planning horizon of the ILP, because this number proved (by trial and error) to be sufficient to obtain consistent results.

At the event "Measurepoint" the number of patients in the system at that moment is registered, per type and in total. The measure points are hourly, and there is a warm-up period and a cool-down period of length 24000 hours. This means that the first 24000 and the last 24000 hours of the simulation do not count towards the confidence interval of the average patients in the system. This confidence interval is calculated using the Batch means method [18].

At the event "Arrival" a patient is admitted at the hospital, and the patients discharge moment ("EndService" event) is planned. If the hospital is full however, the patient is blocked. Aside from this, the next patient of that type is planned.

The arrivals of elective patients are planned according to the strategy found in the ILP, which is extrapolated over the entire simulation run. Their discharge moment is planned with a randomly generated length of stay based on the best fitting probability distribution from the data analysis. The emergency arrivals are planned according to a Poisson process, which means that at each arrival the next arrival is planned with an interarrival time drawn from an exponential distribution. Any other distribution of interarrival times could be implemented here as well. It is important to note here that while the emergency admissions are planned in the ILP, this part of the planning from the ILP is purely to reserve capacity for emergency arrivals, and the intention is not to plan emergency arrivals at these exact moments in time. Moreover, although the patients are assigned to specific beds in the planning, this is not strictly followed in the simulation.

At the "EndService" event, a patient is discharged, so when the "EndService" event happens the counter that keeps track of the number of patients in the system is reduced by one. How the simulation works is further clarified in Figure 1.

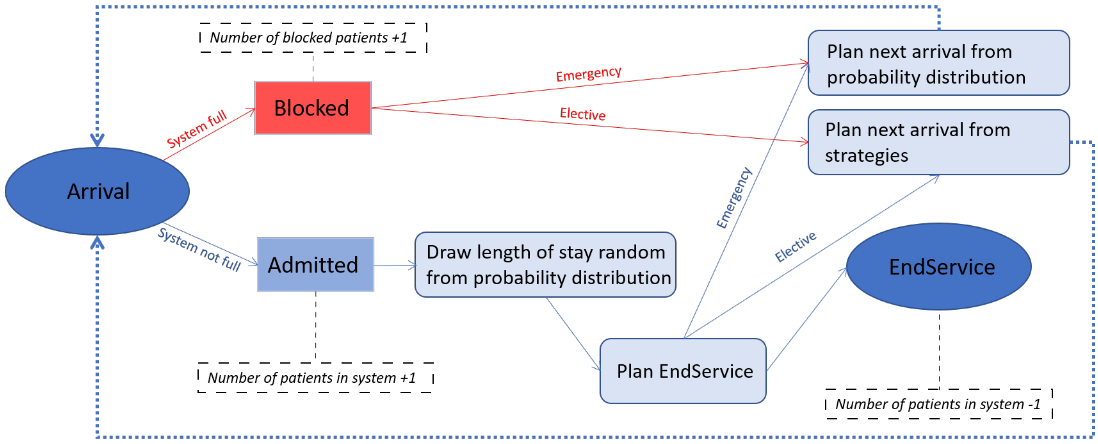

Figure 1: Flow diagram of the Discrete Event Simulation

The simulation tests the strategies from the ILP on blocking probability, which creates the opportunity to test more than one optimal or feasible solution and compare them to each other.
The model was tested in a case study on the VUmc hospital in Amsterdam, which is presented in the next sections.

3 Case study

We applied our model to the situation at the neurology and neurosurgery wards at the VUmc. We had a data set that contained all information of patients that were discharged between 2019-01-01 and 2019-10-10. We selected the data of patients that were admitted between 2019-01-01 and 2019-09-01, since patients that were admitted before 2019-01-01 were only in the data set if they were discharged after 2019-01-01, so using these patients' data would create a less precise data set. And for a similar reason we did not use the last part of the data set. We analyzed the data extensively, but here we only present the main results that are relevant for our model. The full data analysis can be found in Appendix C.
We considered four types of patients: emergency patients with surgery, emergency patient without surgery, elective patients with surgery and elective patients without surgery. Elective and emergency denote the nature of the admission, i.e. emergency patients are unscheduled admissions and elective patients are scheduled admissions. In Table 1 the results on the mean and the standard deviation of the length of stay (LoS) of the different categories of patients are presented.

Table 1: Length of stay data analysis

Type	Surgery	Number of patients	Average LoS	Std.dev LoS
Emergency	Yes	124	207	228
Emergency	No	467	101	123
Emergency	Both	591	123	157
Elective	Yes	246	141	130
Elective	No	243	79	144
All	Both	1080	117	150

From the data analysis we found that an exponential distribution is suitable for the length of stay of all types of patients, and that a Poisson process is suitable for the arrival process of emergency and elective patients. From this we concluded that the Erlang loss model (M/G/c/c queuing model) would be a suitable model for the admission and discharge process at the hospital. This we used to analyze the situation at the hospital with a queuing model to get an initial idea of the required capacity, described in Section 3.1.

3.1 Queuing model analysis

Let λ be the arrival rate (per hour), μ be the average service time (in hours), and c the number of available beds, then the Erlang loss model has the following properties:

1. Blocking probability: The probability that the system is full, so that a patient is blocked, is

$$P_c = \frac{(\lambda\mu)^c/c!}{\sum_{k=0}^{c}(\lambda\mu)^k/k!}$$

2. Average number of occupied beds: The average number of occupied beds is

$$\bar{o} = (1 - P_c)\lambda\mu$$

3. Occupancy rate: And with that the occupancy rate is:

$$r_o = \frac{\bar{o}}{c} * 100\% = \frac{(1 - P_c)\lambda\mu}{c} * 100\%$$

As we found in Table 1 the average length of stay of all patients is 117 hours. The average arrivals per hour is $\frac{\text{Total arrivals}}{\text{Length of data set in hours}} = \frac{1080}{243*24} \approx 0.185$. So $\lambda = 0.185$, $\mu = 117$ and c is the capacity. We investigated the three values as described above for different values of c, of which the results are shown in Table 2.

Table 2: Analysis of the situation at VUmc via a queuing model

c	P_c	$\bar{o}$	r_o
22	14%	19	84%
23	12%	19	83%
24	9.7%	20	81%
25	7.8%	20	80%
26	6.1%	20	78%
27	4.6%	21	76%
28	3.5%	21	75%
29	2.5%	21	73%
30	1.8%	21	71%

3.2 Data analysis via Discrete Event Simulation

We used a Discrete Event Simulation to see at every hour between 2019-01-01 and 2019-09-01 how many patients were present at the hospital wards. This Discrete Event Simulation performs events in chronological order. In this case we used three types of events: "Arrival", "EndService" and "Measurepoint". At an arrival the number of patients in the system increases by one, at the end of service the number of patients in the system decreases by one. The arrivals and service times (in our case the length of stay) are imported from the hospital data. So the simulation shows us exactly at each point in time how many patients were present at the hospital. Using measure points at each hour, we could calculate the average number of occupied beds and average occupancy rate of the hospital, which we display in Table 3. Moreover, we could also calculate the blocking probability by keeping track of the number of patients that are blocked upon arrival because the system is full. Because at the begin and the end of the data set the hospital would be empty in the simulation, we used a so-called warm-up period and cool-down period, which means that we left some results at the start and the end of the simulation out for the statistics. The length of this period was at both the beginning and the end about three weeks (500 hours). In Table 3 we provide a table of the realised blocking probability, average number of occupied beds, and occupancy rate, similar to Table 2.

Table 3: Analysis of the situation of VUmc via Discrete Event simulation

c	P_c	$\bar{o}$	r_o
22	20%	21	96%
23	18%	22	96%
24	16%	23	94%
25	14%	23	94%
26	12%	24	93%
27	10%	25	92%
28	8.3%	26	91%
29	6.5%	26	90%
30	5.1%	27	89%
31	3.3%	27	87%
32	2.3%	27	86%
38	0.0%	28	74%

Comparing Table 3 and Table 2 it is clear to see that the queuing model underestimates the blocking probabilities quite significantly. Therefore we also incorporated the values for $c = 31$ and $c = 32$ in Table 3. On top of that we investigated the actual average number of occupied beds, by setting the capacity at the maximum number of patients in the system over the entire data set, so no patient would be blocked in the simulation. This resulted in an average number of occupied beds of around 28. This is definitely too high for the two wards to be able to merge, especially since the blocking probability given the current admission pattern is not acceptable for 28. So there is a lot of room for improvement in efficiency, since for a 5% blocking probability the queuing model only requires 27 beds, whereas the hospital would require 30 beds.

3.3 ILP input

The ILP was solved using the Python package PULP [13]. For our case study the variables are the following:

- $n_t = 3$, namely $t = 0$ are the emergency patients, $t = 1$ are the elective patients with surgery and $t = 2$ are the elective patients without surgery.
- From Table 1: $l_0 = 123$, $l_1 = 141$, $l_2 = 80$. We did however also try to use an increased LoS in the ILP in some cases to reduce the risk of a patient not being treated in the planned amount of time. This is, where applicable, mentioned in the results tables (Section 4).
- $h = 28$, since this can be easily extended to a planning period of two months, which is the planning period the hospital generally uses.
- The inputs e_d, e_w and e_m differ between the different results, they are mentioned in the results tables (Section 4).
- d_1 and d_2 equal 28, based on the average number of elective patients per day being 1 (Table 1).
- The values of c_{ij}, u_{ik} and v_{im} are determined from the found patterns, using the information we stored in the patterns during the creation, and with functions that can calculate for any pattern the number of emergency arrival spots on any given day, and in any given week.

4 Case study results

As mentioned earlier, for our results we differed the inputs e_d, e_w and e_m and the average length of stay to see which values would give better results. For example, a higher average length of stay results in more beds required, but increases the probability that the planning can be followed. For our initial ILP, we used $e_d = 2$, $e_m = 3$ and $e_w = 17$, since from the data we found that there were on average 2.4 emergency admissions per day, so these numbers should reserve enough capacity for emergency admissions per week. Moreover, we used the average length of stay. A brief calculation shows that with these values we would require at least $123 * 4 * 17 + 141 * 28 + 80 * 28 = 14552$ bed hours, so at least $14552/(24 * 28) \approx 22$ beds. We obtained an optimal solution of 22 beds required from our ILP, so the ILP did result in a solution that is the minimal number of beds possible.
There was actually more than one solution that resulted in 22 beds. For this solution and all other solutions, we used the one that resulted in the lowest blocking probability in the simulation, given the specific input parameters mentioned in Table 4 and Table 5.
In these tables S_n denotes the strategy. All strategies are visualized in Appendix B. $\bar{P}$ denotes the average number of patients in the system over the simulation, for which a 95% confidence interval is provided. C_f denotes the capacity found from the ILP, C_s denotes the capacity used in the simulation. This is different because for instance with S_1, although the ILP resulted in 22 beds, this was found using the average length of stay, and in reality there will be patients that need to be treated longer than this average amount, which also happens in the simulation. Moreover, in the simulation emergency patients arrive according to a Poisson process, so they might arrive when the system is full. Therefore we used different (larger) capacities in the simulation then found in the ILP.
Because of the average number of admissions being 2.4 per day, we used a mean interarrival

time of 10 hours for the emergency arrivals in the simulation. In the tables we present the results with a blocking probability around 5%, other results we obtained can be found in Appendix A.

4.1 Results based on found probability distributions

Table 4: Results simulation using the DES based on found probability distributions
* 10% more elective patients (respectively 3 with and 3 without surgery extra)

S_n	LoS	e_d	e_m	e_w	C_f	C_s	95%CI($\bar{P}$)	P_c
S_1	Average	2	3	17	22	26	(17, 23)	5.4%
S_2	Average	2	3	17	22	27	(17, 24)	3.9%
S_3	+10%	2	3	17	25	26	(17, 23)	5.9%
S_3	+10%	2	3	17	25	27	(17, 24)	4.3%
S_4	Average	2	3	19	24	26	(17, 23)	5.6%
S_4	Average	2	3	19	24	27	(17, 24)	4.1%
S_5	Average	3	3	21	26	26	(17, 23)	6.0%
S_6	Average	3	3	21	26	27	(18, 24)	4.7%
S_8*	Average	2	3	17	23	27	(18, 24)	5.7%
S_8*	Average	2	3	17	23	28	(18, 25)	4.3%

As can be seen from Table 4, the strategy that resulted in the lowest blocking probabilities was for both cases (capacity 26 or 27) the strategy found using the average length of stay, a minimal of two and a maximum of three emergency arrivals per day and a minimum of 17 emergency arrivals per week as input for the ILP, although the difference with the other approaches is not too large.
Furthermore, these results show that using a capacity of 27 results in a blocking probability vast below 5%, and using a capacity of 26 results in a blocking probability between 5% and 6%. Aside from this the results in Table 4 show that admitting 10% more elective patients results in one more bed required to obtain a similar blocking probability, which is logical considering 10% more elective patients means that on a planning horizon of 28 days, there are three more elective patients with surgery and three more elective patients without surgery to be treated. This equals $(3 * 141 + 3 * 80)/24 \approx 27.6$ "bed days", or $27.6 * 24$ "bed hours", required, which fills up exactly one bed in the strategy.

4.2 Results based on hospital data

To test the strategies more extensively, we did not only run a simulation with approximate probability distributions, but we also implemented a simulation that uses hospital data. We used the same methodology as in Table 3 with the only difference that in this simulation the elective patients are not taken from the data but are planned according to the strategy. Their length of stay is drawn randomly from a list of all lengths of stay of the hospital data of elective patients. To clarify this modification of the simulation Figure 2 visualizes this version of the simulation.

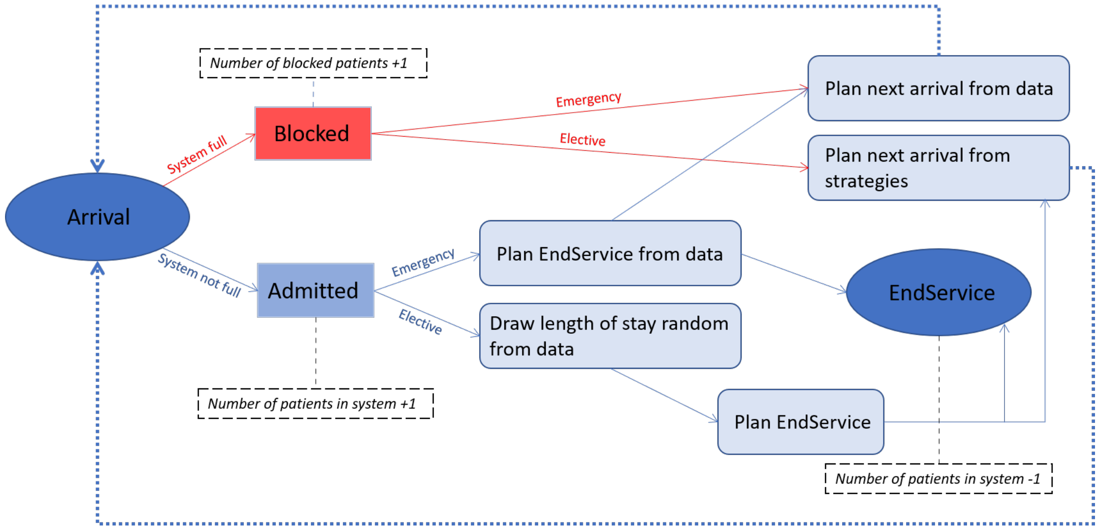

Figure 2: Flow diagram of the Discrete Event Simulation

All results are calculated in the same way as in Table 3 with one exception. We ran the simulation 100000 times and took the average values. We did this because we randomly drew the length of stay of the elective patients, and this influences the results, so we required more simulation runs to be certain we had consistent results. Since Table 5 is intended to show a comparison between our strategy and the strategy of the hospital, the values displayed in Table 3 are also shown in Table 5. Moreover, we show the improvement in blocking probability of our strategy over the hospital's strategy when using the same capacity.

Table 5: Results simulation using the DES based on found probability distributions * 10% more elective patients (respectively 3 with and 3 without surgery extra), the improvement here is still taken against the original hospital data

S_n	LoS	e_d	e_m	e_w	C_f	C_s	P_c	$\bar{o}$	r_o	Improvement
S_{10}	Average	2	3	17	23	28	6.2%	24	84%	25%
S_{11}	Average	2	3	17	22	29	4.4%	24	82%	32%
S_0	Average	2	3	17	22	30	2.7%	24	80%	47%
S_{13}	+10%	2	3	17	25	28	5.8%	23	84%	30%
S_{14}	+10%	2	3	17	24	29	4.3%	24	84%	34%
S_{14}	+10%	2	3	17	24	30	2.5%	24	81%	51%
$S_{15}*$	Average	2	3	17	23	29	5.2%	24	83%	20%
S_8*	Average	2	3	17	23	30	4.1%	25	82%	20%
S_{17}	Average	2	3	19	24	28	5.8%	23	83%	30%
S_{18}	Average	2	3	19	24	29	5.1%	24	83%	22%
S_{19}	Average	2	3	19	24	30	3.7%	24	81%	27%
S_{20}	Average	3	3	21	26	28	6.6%	24	84%	20%
S_6	Average	3	3	21	26	29	5.0%	24	82%	23%
S_{21}	Average	3	3	21	26	30	2.9%	24	81%	43%

The results show that our method improved the current blocking probability at the hospital by 20% up to 51%. The most important cases with a blocking probability around 5% (capacity of 28 or 29 beds) show an average of 27% improvement over the respective 8.3%

and 6.5% blocking probability the hospital would have achieved with these capacities.
The best results for these capacities were, contrary to Table 4, achieved using 110% of the average length of stay as input to the ILP. Interesting to note is that a capacity of 29 is not only sufficient for the current situation using this strategy (if accepting a blocking probability of 5%), but is also almost sufficient for 10% more elective patients.
The results in Table 5 show that in this simulation an average of two beds more was required to achieve a similar blocking probability as in Table 4, which is due to the difference between the actual hospital data and the probability distributions that were chosen to best fit that data, mainly the fact that the emergency arrivals are not as evenly spread over the day and the week as a Poisson distribution with constant arrival rate assumes.

5 Conclusion

In our research we created a model to optimize the scheduling strategy at a hospital, taking into account the unscheduled admissions while planning the scheduled admissions. We modified the cutting stock problem to obtain scheduling strategies, and tested these strategies in two simulations. To conclude our results, we can safely state that the hospital, using our planning strategy, needs 28 beds on one ward to be able to merge the two wards. Moreover, the hospital would need one extra bed to admit 10% more elective patients.
Based on the data analysis, and the calculation of 22 beds needed in an ideal situation, it was expected that the currently used 30 beds could be improved on. Based on our results, 28 beds is achievable using our obtained strategies, which is a significant improvement on the current situation.
The strategy that resulted in the lowest blocking probability for 28 beds was found using either a minimum of three emergency admissions per day as input for the ILP, or an increased length of stay as input for the ILP, instead of the basic input (used for example for strategy 0). Based on the results of the other simulation, and on the results for other capacities, we would advice to use an increased length of stay (by 10%), a minimum of two and a maximum of three reserved admission spots for emergency admissions per day, and 17 admissions spots per week for emergency admissions as input for the ILP. The strategy we would advice the hospital to use for 28 beds, would be strategy 13 (see Appendix B).
We used a modified version of the cutting stock problem to optimize the planning strategy. One downside to this linear program model is the finite planning horizon, which we solved by using a dynamic planning. Another downside to this choice of linear program model is that it can not take into account randomness of length of stay. We solved this problem partially by overestimating the length of stay of patients, but this also creates more wasted "bed hours" if a patients is actually discharged earlier than expected. A possible solution to this problem would be to categorize the patients into more categories to create lower standard deviation of the length of stay for each category.

6 Further research

For further research we would recommend considering more different inputs to the ILP, such as a combination of an increased length of stay and more reserved admission spots for emergency patients per day or per week. Moreover, what could be improved upon in our model is taking into account that emergency admissions are not necessarily randomly divided over the day or over the week, which is confirmed by our data analysis (Appendix C). One could also consider more types of elective or emergency patients, to possibly make the standard deviation of the length of stay smaller. This would result in the strategy being more likely to be followed as intended. In case of the emergency patients, however, considering more types is not realistic, since it is not possible to know before arrival what type the emergency patient is. Finally, an improvement that could be made is to test the strategies on a larger data set.

7 List of symbols

Table 6: Caption

	Description	Unit	Page
h	Planning horizon	days	3
t	Patient type		3
n_t	Number of patient types		3
l_t	Required bed hours for patient of type t	hours	4
N_{it}	Number of patients of type t in pattern i		4
x_i	Decision variables ILP		5
d_t	Number of patients of type t to be treated within the planning horizon		5
e_d	Minimum emergency arrivals per day		5
e_m	Maximum emergency arrivals per day		5
e_w	Minimum emergency arrivals per week		5
c_{ij}	Number of patients of type t in pattern i		5
u_{ij}	Number of emergency spots on day j in pattern i		5
v_{ij}	Number of emergency spots in week j in pattern i		5
LoS	Length of stay	hours	7
λ	Queuing model arrival rate	1/hour	7
μ	Queuing model average service time	hours	7
c	Queuing model capacity		7
P_c	Blocking probability		7
$\bar{o}$	Average occupied beds		7
r_o	Occupancy rate		8
S_k	Strategy number k		10
$\bar{P}$	Average patients in system		10
C_f	ILP output capacity		10
C_s	Simulation input capacity		10

References

[1] T.J. Bailey. A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society: Series B.*

[2] Leonardo S.L. Bastos, Janaina F. Marchesi, Silvio Hamacher, and Julia L. Fleck. A mixed integer programming approach to the patient admission scheduling problem. *European Journal of Operational Research*, 273(3):831 – 840, 2019.

[3] Dimitris Bertsimas. An analytic approach to a general class of g/g/c queuing systems. 1987.

[4] A.M. de Bruin, R. Bekker, L. van Zanten, and G.M. Koole. Dimensioning hospital wards using the erlang loss model. *Annals of Operations Research*, 178:23–43, 2010.

[5] Maxence Delorme, Manuel Iori, and Silvano Martello. Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1):1 – 20, 2016.

[6] F. Gorunescu, S.I. McClean, and P.H. Millard. A queuing model for bed-occupancy management and planning of hospitals. 2002. J Oper Res Soc, 53 (1), pp.19-24.

[7] F. Gorunescu, S.I. McClean, and P.H. Millard. Using a queuing model to help plan bed allocation in a department of geriatric medicine. 2002. Health Care Manage Sci, 5, pp.307-312.

[8] C. Granja, B. Almada-Lobo, F. Janela, J. Seabra, and A. Mendes. An optimization based on simulation approach to the patient admission scheduling problem using a linear programming algorithm. *Journal of Biomedical Informatics*, 52:427–437, 2014. cited By 14.

[9] M. Harzi, J.-F. Condotta, I. Nouaouri, and S. Krichen. Using the hybrid ils/vnd method for solving the patients scheduling problem in emergency department: A case study. volume 126, pages 733–742, 2018. cited By 0.

[10] Saif Kifah and Salwani Abdullah. An adaptive non-linear great deluge algorithm for the patient-admission problem. *Information Sciences*, 295:573 – 585, 2015.

[11] A.G. Leeftink, I.M.H. Vliegen, and E.W. Hans. Stochastic integer programming for multi-disciplinary outpatient clinic planning. *Health Care Management Science*, 22(1):53–67, 2019. cited By 3.

[12] X. Li, P. Beullens, D. Jones, and M. Tamiz. An integrated queuing and multi-objective bed allocation model with application to a hospital in china. 2009. J Oper Res Soc, 60, pp.330-338.

[13] Anita Mason Andrew O'Sullivan Michael Philips Antony Mitchell, Stuart Kean. Optimization with pulp, 2009. https://pythonhosted.org/PuLP/.

[14] L.P. Seelen. An algorithm for ph/ph/c queues. 1986. European Journal of Operations Research 23, pp.118-127.

[15] Y. Takahashi. A lumping method for numerical calculations of stationary distributions of markov chains. 1975. Research Report B-18, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan.

[16] Y. Takahashi and Y. Takami. A numerical method for the steady-state probabilities of gi/g/c queuing systems in a general class. 1976. *Journal of the Operational Research Society of Japan* 19 147-157.

[17] Aykut Melih Turhan and Bilge Bilgen. Mixed integer programming based heuristics for the patient admission scheduling problem. *Computers Operations Research*, 80:38 – 49, 2017.

[18] Unknown. Batch means method. http://www.netlab.tkk.fi/opetus/s38148/s02/luennot /E_statanal.pdf.

[19] Unknown. Phase-type distribution. https://en.wikipedia.org/wiki/Phase-type_distribution.

[20] Noëlle Maria van de Vrugt. *Efficient healthcare logistics with a human touch*. PhD thesis, University of Twente, Netherlands, 7 2016.

[21] J.S. Wu. Refining the diffusion approximation for the g/g/c queue. *Computers and Mathematics with Applications*, 20(11):31–36, 1990. cited By 3.

[22] J.P. Young. Stabilization of inpatient bed occupancy through control of admissions. *Journal of the American Hospital Association*, 39:41–48, 1965.

Appendix

Appendix A: complete results

In this Appendix all documented results are presented.

Results of simulation based on found probability distributions

Table 7: Results simulation using the DES based on found probability distributions
* 10% more elective patients (respectively 3 with and 3 without surgery extra)

S_n	LoS	e_d	e_m	e_w	C_f	C_s	95%CI($\bar{P}$)	P_c
S_0	Average	2	3	17	22	25	(17.12, 22.50)	7.29%
S_1	Average	2	3	17	22	26	(17.25, 23.18)	5.38%
S_2	Average	2	3	17	22	27	(17.18, 23.89)	3.90%
S_3	+10%	2	3	17	25	25	(17.34, 22.55)	7.30%
S_3	+10%	2	3	17	25	26	(17.26, 23.34)	5.85%
S_3	+10%	2	3	17	25	27	(17.43, 23.83)	4.25%
S_4	Average	2	3	19	24	25	(17.44, 22.73)	7.59%
S_4	Average	2	3	19	24	26	(17.44, 23.42)	5.64%
S_4	Average	2	3	19	24	27	(17.49, 23.84)	4.14%
S_5	Average	3	3	21	26	26	(17.28, 23.25)	6.02%
S_6	Average	3	3	21	26	27	(17.73, 24.02)	4.74%
S_7*	Average	2	3	17	23	25	(17.84, 22.69)	9.35%
S_8*	Average	2	3	17	23	26	(18.06, 23.64)	7.32%
S_8*	Average	2	3	17	23	27	(18.11, 24.32)	5.69%
S_8*	Average	2	3	17	23	28	(18.34, 24.78)	4.27%

Results of simulation based on hospital data

Table 8: Results simulation using the DES based on found probability distributions * 10% more elective patients (respectively 3 with and 3 without surgery extra), the improvement here is still taken against the original hospital data

S_n	LoS	e_d	e_m	e_w	C_f	C_s	P_c	$\bar{o}$	r_o
S_1	Average	2	3	17	22	26	9.31%	22.17	85.27%
S_9	Average	2	3	17	22	27	7.84%	22.80	84.44%
S_{10}	Average	2	3	17	23	28	6.23%	23.65	84.45%
S_{11}	Average	2	3	17	22	29	4.38%	23.81	82.10%
S_0	Average	2	3	17	22	30	2.69%	24.12	80.40%
S_{12}	+10%	2	3	17	25	26	9.04%	21.95	84.42%
S_3	+10%	2	3	17	25	27	8.23%	22.93	84.93%
S_{13}	+10%	2	3	17	25	28	5.81%	23.38	83.50%
S_{14}	+10%	2	3	17	24	29	4.31%	24.34	83.93%
S_{14}	+10%	2	3	17	24	30	2.47%	24.27	80.90%
S_8*	Average	2	3	17	23	27	9.07%	22.98	85.11%
$S_{15}*$	Average	2	3	17	23	28	7.72%	23.75	84.82%
$S_{15}*$	Average	2	3	17	23	29	5.24%	24.14	83.24%
S_8*	Average	2	3	17	23	30	4.09%	24.68	82.27%
S_4	Average	2	3	19	24	26	10.43%	22.26	85.62%
S_{16}	Average	2	3	19	24	27	7.56%	22.64	83.85%
S_{17}	Average	2	3	19	24	28	5.82%	23.22	82.93%
S_{18}	Average	2	3	19	24	29	5.09%	24.07	83.00%
S_{19}	Average	2	3	19	24	30	3.67%	24.24	80.80%
S_{20}	Average	3	3	21	26	26	11.57%	22.23	85.50%
S_{20}	Average	3	3	21	26	27	8.45%	22.45	83.15%
S_{20}	Average	3	3	21	26	28	6.61%	23.51	83.96%
S_6	Average	3	3	21	26	29	5.02%	23.73	81.83%
S_{21}	Average	3	3	21	26	30	2.93%	24.21	80.70%

Appendix B: strategies

The strategies that are shown here are the strategies that led to the results found in Table 4 and Table 5. Day 0 is the first Monday.
Green denotes an elective with surgery admission, yellow an elective without surgery.
The numbers in the table represent the planned time of admission.
The emergency arrivals are omitted on purpose, since they are not actually planned in reality, as explained in Section 4.

Strategy 0:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	8	8	16	8	8			16	13	8	8	8			16	8	14	8	8			8	11	8	13	8		
	8	8	16	16	16				13	8	8	14				8	17		16			8	14		14			
	8	8		16	16					8						8			16			16	14		16			
	8			16	16					11						8			17						16			
	8															8												
	8																											
	8																											

Strategy 1:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	8	8	8	16	8				8	11	8	14			8	8	8	11	8			8	14	8	16	8		
	8	8	8		8				8	13	8	14			8	9	8	11	9			11		8		8		
	16	8			16				8			16				16		16	9			17		8		8		
	16				16				8			16										17		9				
	16				16				8			16																

Strategy 2:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	8	8	8	8	8			8	8	8	8	8			8	8	8	9	8			8	8		11	8		
	8	8	8		16			8	8			8			8	14		16	8			14	12		11	8		
	8		8						8						10	17		16	8			14	17		11	8		
	8		16						8						16	17			16							11		
			16						13						16											14		

Strategy 3:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	8	8	8	16	8			8	8	8		8			8	8	18	15	8			8	8	8	10	8		
	8	16		16	8			8	8	15		8				8	18		8			9	8	8		8		
	8			16	16				8			14				8			8			9		10		8		
					16				15			14				14			14					15		8		
					16				15			15				18			15							8		

Figure 3: Strategy 0 up to and including 3

Strategy 4:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	8	8	8	8	8			8	8	8	11	8			8	9	8	16	8			11	8	9	11	8		
	8			8	8			16	8	8	16	8			8	9			8			11	8	13	14	8		
	8			16	8			16		13		8			8				9			17			14			
	9			16	16					13		8			14				17									
	9																		17									
	16																		17									

Strategy 5:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	8	8	8	8	8			8	8	8	8	11			8	16	8	8	8			8		8	11	14		
	8	8			8			8	8	11	8	16			8	16		13	8			8		9	17			
	8				16			16	8		13	16			8	16		16	8			9		10				
	8				16										14							12						
	16				16										14							13						
					16																	13						
					16																							
					16																							

Strategy 6:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	8	8	8	16	8			8	8	11	8	8			8	8	8	16	8			8	8	8	8	16		
	8	8		16	16				8	11	14				8	8		17	8			9	8	8	16	16		
	8	8		16	16				8	13	14				8	16							13	13	16			
	16			16	16					13					16	16							13					
					16										16													
															16													

Figure 4: Strategy 4 up to and including 6

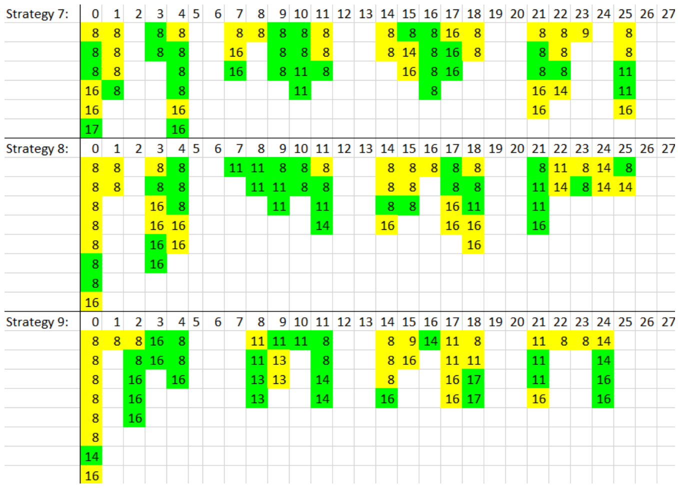

Figure 5: Strategy 7 up to and including 9

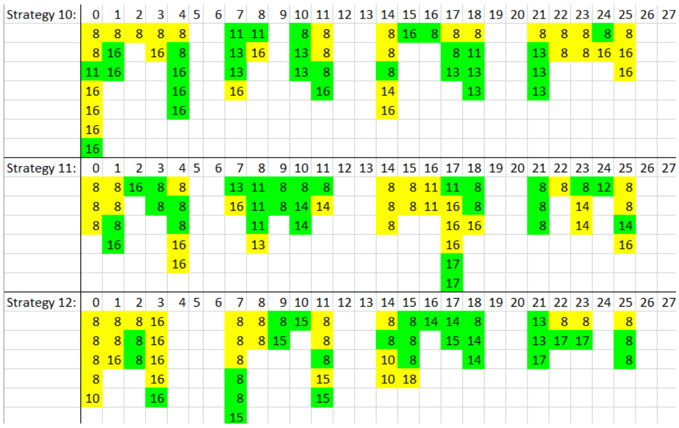

Figure 6: Strategy 10 up to and including 12

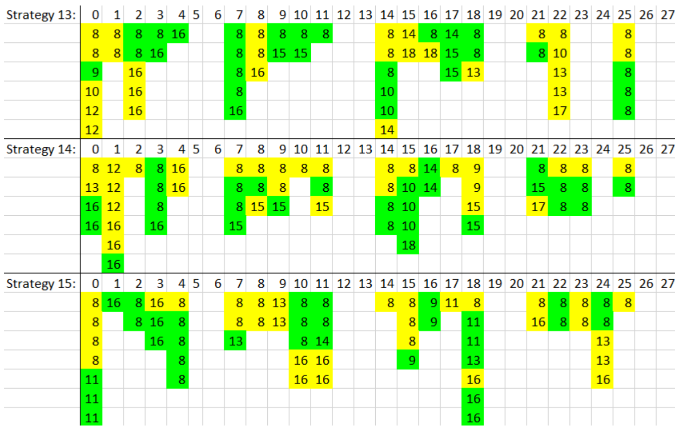

Figure 7: Strategy 13 up to and including 15

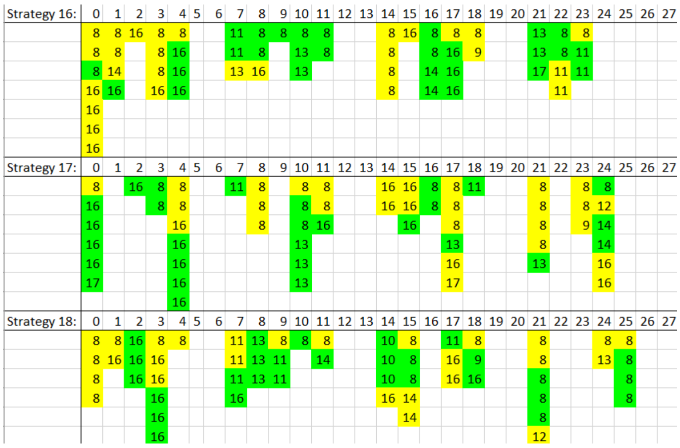

Figure 8: Strategy 16 up to and including 18

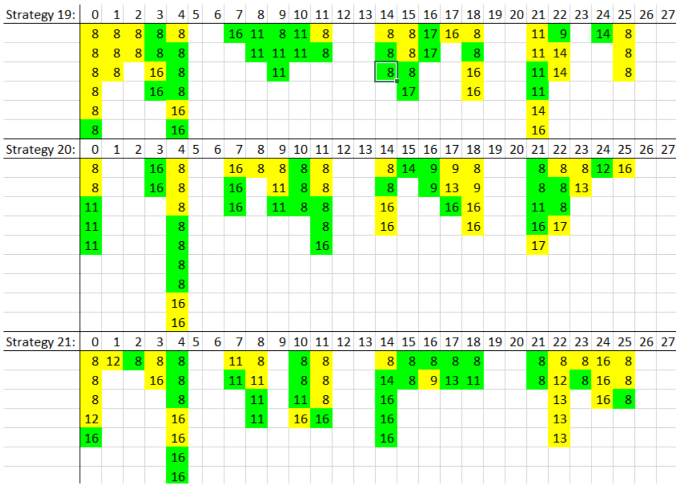

Figure 9: Strategy 19 up to and including 21

Appendix C: Case study data analysis

We analyzed the data using R, Excel and Python. In R we modified the data, i.e. removed the data that was not necessary for our research. Afterwards we used both R and Excel to create Figures to visualize the data, and we used R to perform the statistical tests.
We first analyzed the length of stay of the patients, and afterwards the admission pattern. With those two combined we set up a queuing model to analyze the blocking probabilities of the system for given capacities. Finally, we used a Discrete Event Simulation in Python to analyze the data, which we describe in the corresponding section.

The hospital data contained information of all patients that were discharged between 2019-01-01 and 2019-10-10. This resulted in some issues, since some patients that were admitted in 2018 were in the data set, and some (who were discharged before 2019, i.e. with a shorter length of stay) were not. Moreover, patients who were admitted in October 2019 but not discharged before 2019-10-10 were not in the data set, while patients admitted at the same moment but discharged earlier were in the data set. Therefore we chose to only consider for our data analysis patients who were admitted between 2019-01-01 and 2019-09-01. Since for these patients, apart from possibly a negligible number of exceptions, they were all discharged before 2019-10-10. So our data sets contains full information on all patients admitted in this period.

Length of stay analysis

The results on the average length of stay (LoS) and the standard deviations of the length of stay are shown in Table 9. The length of stay is mentioned in hours.

Table 9: Length of stay data analysis

Type	Surgery	Number of patients	Average LoS	Std.dev LoS
Emergency	Yes	124	206.763	228.3737
Emergency	No	467	100.5051	122.9358
Emergency	Both	591	122.7995	157.1141
Elective	Yes	246	140.6069	130.028
Elective	No	243	79.05604	143.9958
Both	Both	1080	117.0134	149.8563

It is clear from these results that there is a major difference between the average length of stay of patients that had surgery and patients that did not have surgery. Therefore it is important to consider whether it is possible to have a distinction between these two in the model. For elective admissions this should be possible, because we may assume that it is known up front whether they need surgery, in at least most of the cases. For emergency patients this might be different however. On top of the difference between patients with and without surgery, there is also a large difference between the average length of stay of emergency patients and elective patients.

Best fitting probability distribution(s)

Since for exponential distributions the mean is the same as the standard deviation, for the emergency patients and the elective admissions with surgery we checked whether an exponential distribution could be a good fit. In Figure 10 the length of stay is shown for emergency patients with and without surgery, and all emergency patients together. Especially for the non-surgery patients and all emergency patients combined, this resembles an exponential distribution. We used a QQ-plot to visualize whether the emergency patients with surgery resembled an exponential distribution with mean $1/206.763 = 0.0048464553$, the emergency patients without surgery resembled an exponential distribution with mean $1/100.5051 = 0.0099497438$ and all emergency patients combined resembled an exponential distribution with mean $1/122.7995 = 0.0081433556$. These are shown in Figures 11 and 12. In Figure 11 it is also visualized that the exponential distribution resembles the shape of the length of stay very accurately if we remove the couple of outliers with a length of stay higher than 600 hours. Also for Figure 12 we removed the couple (respectively six for patients with surgery and two for patients without surgery) extreme outliers.

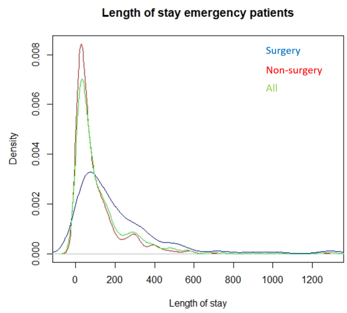

Figure 10: Length of stay of emergency patients

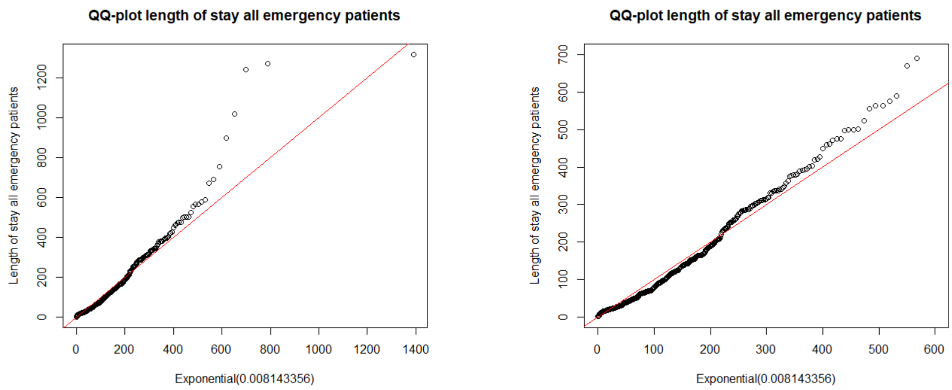

Figure 11: QQ-plots of length of stay of all emergency patients with (left) and without (right) extreme outliers against an exponential distribution

Figure 12: QQ-plots of length of stay of emergency patients with surgery (left) and without surgery (right) against an exponential distribution

From these Figures we conclude that the exponential distribution is an accurate fit for the distribution of the length of stay for emergency patients, only the outliers are more extreme than an exponential distribution would suggest. Moreover, we conclude that if the hospital does not know up front whether an emergency patients needs surgery or not, the hospital can, at least for the length of stay, consider them as one category, with the length of stay Exponential(0.008143356) distributed.
For the elective admissions we do consider the patients with surgery and without surgery to be a different category. As is visible in Figure 13, there is a major difference between the elective patients with and without surgery. And just as the emergency patients they both resemble an exponential distribution. Therefore we made a QQ-plot for both of these length of stay distributions as well, which are shown in Figure 14. Again the extreme outliers (length of stay higher than 600 hours) are removed (respectively five for elective patients with surgery and seven for elective patients without surgery).

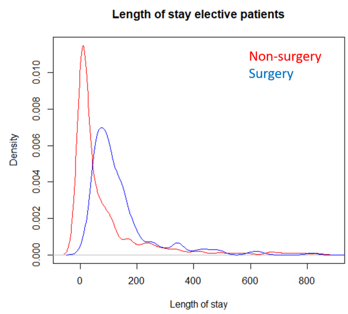

Figure 13: Length of stay of all elective patients

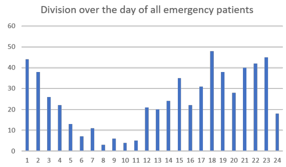

Figure 16: Division over the day of emergency patient admissions

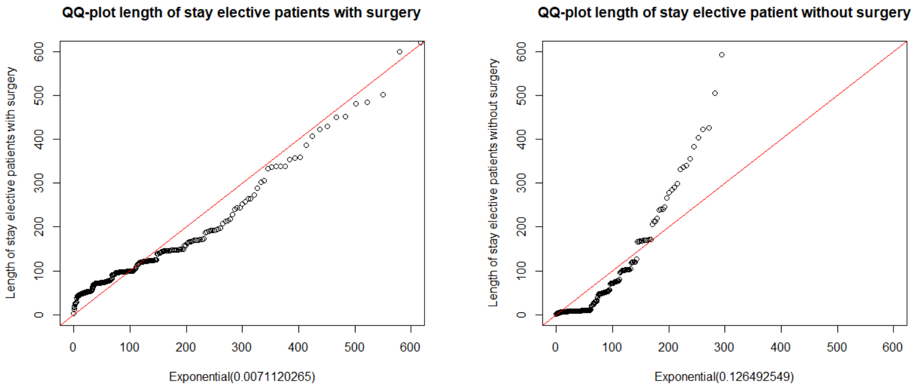

Figure 14: QQ-plot of the length of stay of elective patients with surgery (left) and without surgery (right) against an exponential distribution

It is clear that the exponential distribution is not a good fit for the elective patients without surgery, which was already clear from Table 9, since the standard deviation and mean were far apart, whereas those are the same for an exponential distribution.

Admission pattern analysis

The admission pattern of emergency patients is of significant importance for our optimization model. The division of emergency admissions over the week is visualized in Figure 15.

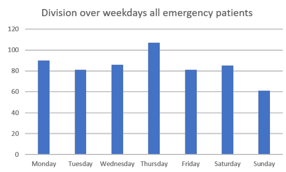

Figure 15: Division of emergency admissions over the weekdays

The graph shows that, aside from Sunday, the emergency admissions are well-distributed over the week.
Figure 16 shows the division of patients over the hours on a day. From this it is clear that the patient admissions are not equally divided over the hours of a day, and therefore a phase-type distribution could well be the best fitting probability distribution.

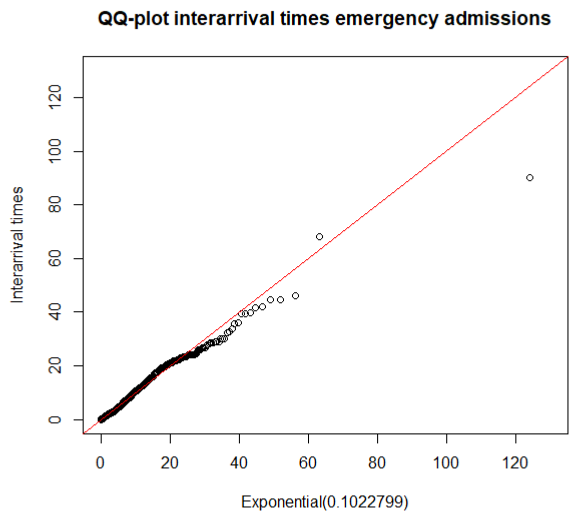

Figure 17: QQ-plot interarrival times emergency admissions against an exponential distribution

Best fitting probability distribution(s)

As described in [22], a Poisson distribution is realistic for unscheduled admissions. Therefore we investigated if the interarrival times of the emergency admissions in our data set are indeed exponential. The results are visualized in a QQ-plot in Figure 17

Since for the queuing model we also needed a distribution for the elective admissions, we also investigated if the current elective admission pattern was a Poisson process by plotting the interarrival times against an exponential distribution. This is shown in figure 18.

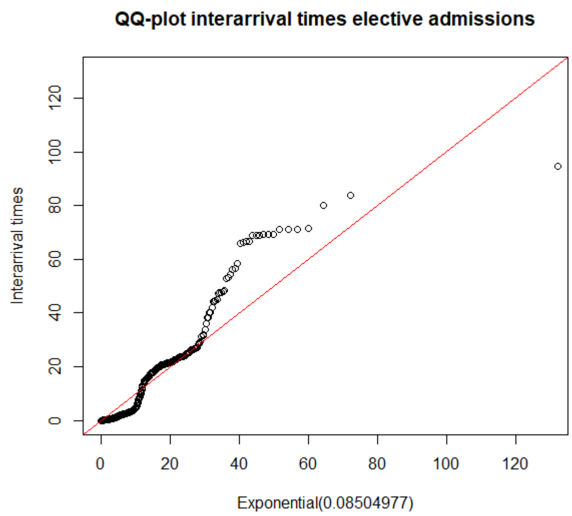

Figure 18: QQ-plot interarrival times elective admissions against an exponential distribution

It is clear from this figure that the interarrival times of elective admissions are not described well by an exponential distribution. However, for our queuing model it would be useful if we could put all types of patients onto one big pile. Therefore we investigated the interarrival times of all patient admissions together, which is visualized in Figure 19.

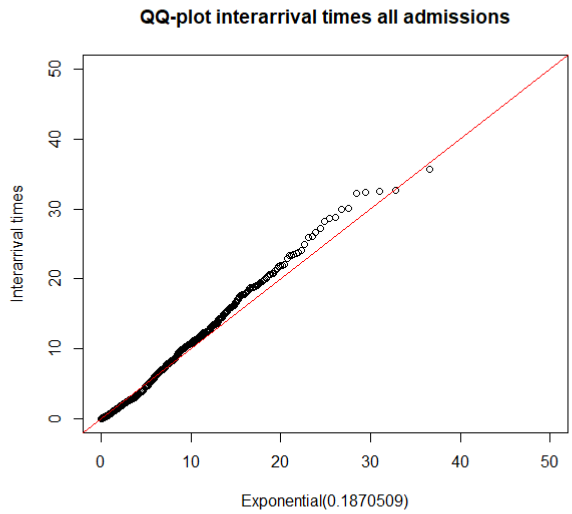

Figure 19: QQ-plot interarrival times all patient admissions

These interarrival times do resemble an exponential distribution, so if we create a queuing model where we take all patients as one type, the Poisson distribution is a good fit for the arrival process.

Best fitting queuing model

The best fitting queuing model given the probability distributions that is also reasonably simple is the Erlang loss model, or in Kendall's notation the M/G/c/c queue. Since the Erlang loss model only requires a mean and a standard deviation of the length of stay, the problem of the exponential distribution not being a good fit for the length of stay for each type of patient is not a problem anymore. Moreover, a Poisson process has been shown to be a good fit for the arrival process of all patient admissions together. Since putting all patients together gives us a reasonable estimate of the necessary capacity, we used the M/G/c/c queue for the blocking probability analysis. One could look into the Phase type distribution for the arrival process, or into separating the different types of patients, but for simplicity reasons we did not consider this here.

Queuing model analysis

Let λ be the arrival rate (per hour), μ be the average service time (in hours), and c the number of available beds, then the Erlang loss model has the following properties:

1. Blocking probability: The probability that the system is full, so that a patient is blocked, is

$$P_c = \frac{(\lambda\mu)^c/c!}{\sum_{k=0}^{c}(\lambda\mu)^k/k!} \tag{1}$$

2. Average number of occupied beds: The average number of occupied beds is

$$\bar{o} = (1 - P_c)\lambda\mu \tag{2}$$

3. Occupancy rate: And with that the occupancy rate is:

$$r_o = \frac{\bar{o}}{c} * 100\% = \frac{(1 - P_c)\lambda\mu}{c} * 100\% \tag{3}$$

As we found in Table 9 the average length of stay of all patients is 117.0134 hours. The average arrivals per hour is $\frac{\text{Total arrivals}}{\text{Length of data set in hours}} = \frac{1080}{243*24} \approx 0.185$. So $\lambda = 0.185$, $\mu = 117.0134$ and c is the capacity. We investigated the three values as described above for different values of c, of which the results are shown in Table 10.

Table 10: Analysis of the situation at VUMC via a queuing model

c	P_c	$\bar{o}$	r_o
22	14.41%	19	84.22%
23	11.95%	19	82.88%
24	9.73%	20	81.43%
25	7.77%	20	79.86%
26	6.07%	20	78.20%
27	4.64%	21	76.45%
28	3.47%	21	74.63%
29	2.52%	21	72.76%
30	1.79%	21	70.87%

We show the results for c between 22 and 30, since a blocking probability higher than 15% is clearly not desirable, and 30 as an upper bound because that is the current maximum capacity at the VUMC, and gives a very acceptable blocking probability of under 2%. From these results we concluded that 26 or 27 beds should be achievable with an optimized planning strategy. These results are useful but required a lot of assumptions, therefore we also investigated the blocking probabilities for certain capacities given our data, i.e. the percentage of patients blocked if a certain capacity was used between 2019-01-01 and 2019-09-01. This we did via a Discrete Event simulation, which is described in section 7.

Data analysis via Discrete Event Simulation

With a Discrete Event Simulation we could see at every hour between 2019-01-01 and 2019-09-01 how many patients were present at the hospital departments. What a Discrete Event Simulation does, is performing events in chronological order. In our case we used three types of events: “Arrival”,“EndService” and“Measurepoint”. At an arrival the number of patients in the system increases by one, at the end of service the number of patients in the system decreases by one. The arrivals and service times (in our case the length of stay) are imported from the hospital data. So the simulation shows us exactly at each point in time how many patients were present at the hospital. Using measure points at each hour, we could calculate the blocking probability, average number of occupied beds and average occupancy rate of the hospital, which we display in Table 11. Because of incompleteness of data at the begin and end of the data set, as discussed at the start of this chapter, we used a so-called warm-up period and cool-down period, which means that we left some results at the start and the end of the simulation out for the statistics. The length of this period was at both the beginning and the end about three weeks (500 hours). In Table 11 we provide a table of the realised blocking probability, average number of occupied beds, and occupancy rate, similar to Table 10.

Table 11: Analysis of the situation of VUMC via Discrete Event simulation

c	P_c	$\bar{o}$	r_o
22	20.37%	21	96.19%
23	18.06%	22	95.61%
24	16.30%	23	94.29%
25	13.70%	23	93.97%
26	11.85%	24	93.30%
27	10.09%	25	92.47%
28	8.33%	26	91.37%
29	6.48%	26	90.06%
30	5.09%	27	88.64%
31	3.33%	27	87.12%
32	2.31%	27	85.55%
38	0.00%	28	73.51%

Comparing Table 11 and Table 10 it is clear to see that the queuing model underestimated the blocking probabilities quite a bit. Therefore we also incorporated the values for $c = 31$ and $c = 32$. On top of that we investigated the actual average number of occupied beds, by setting the capacity at the maximum number of patients in the system over the entire data set. This resulted in an average number of occupied beds of around 28. This is definitely too high for the two departments to be able to merge, especially since the blocking probability given the current admission pattern is not acceptable for 28. So there is a lot of room for improvement in efficiency, since for a 5% blocking probability the queuing model only needs 27 whereas, the hospital would have needed 30 beds.