MASTER

State based patient scheduling with a scare healthcare server and beds in a finite priority queue

Wolvers, R.

Award date:
2021

State based patient scheduling with a scarce healthcare server and beds in a finite priority queue

Richard Wolvers
Student identity number (0894489)

Supervisors:
ir. dr. S.D.P Flapper
dr. ir. N. P. Dellaert

In partial fulfilment of the requirements for the degree of
Master Of Science
in Operations Management and Logistics

Eindhoven University of Technology
Eindhoven
December 17, 2020

Keywords: MDP, uniformization, Patient scheduling policy, Triage, Deteriorating patients, Abandoning patients

Abstract

In this study, the problem of scheduling deteriorating patients with a scarce healthcare server in a finite queue is tackled. This problem was motivated by the 2020 Covid-19 pandemic outbreak and the related issues around intensive care treatment. Some practical issues exist with the currently used method that encompasses always treating the most serious patient first. Only in very dire situations (e.g war or pandemic), the least serious first would be prioritized. In this study, the problem is modeled as a Continuous-Time Markov Decision Process (CTMDP). Two different situations are considered. One without internal recovery in the healthcare department and one with internal recovery in the healthcare department. The uniformization method is used to convert the CTMDP into a Discrete Time Markov Decision Process (DTMDP) model, solved with Matlab. Different scenarios are proposed: 1 server and 2 priority classes, 3 servers and 2 priority classes, and 1 server and 3 priority classes. Different cases per scenario are analyzed with most cases resulting in a 'switching curve' policy, where the switching curve splits the optimal scheduling policy into two action regions (Treat class 1 patient or class 2 patient). Different parameter values show different switching curves (e.g. arrival rates). This shows that a state based scheduling policy should change depending on the arrival rate (if known beforehand). Comparing the optimal scheduling policy with other simple policies shows the importance of using a state-based policy. Moreover, the results indicate that it is important to not only consider the patient mix waiting for treatment, but also the patient mix in treatment and in recovery, while scheduling waiting patients.

Summary

In 2020 the sudden uprise of the Covid-19 (corona disease) pandemic has put a lot of pressure on Intensive Care Units (ICU) worldwide. Several challenges arose with one of the challenges being how to deal with a surge of patients, while there is only a small amount of resources available to treat patients. When a healthcare system with scarce resources is faced with a queue of patients, there are three options for improving the current situation: Increase the number of resources, regulate the queue by deciding which patient to accept into the queue or change the current policy of deciding which patient should be treated first. The first option can be very costly, as well as requiring a lot of time to realize, whereas the second and third option do not require any extra costs, except for a possible investment in a decision making system. This thesis focuses on the third option. This problem does not only occur at the Intensive Care (IC) but can also occur at the Emergency Department (ED) or for ambulance transportation. In practice, patients are always scheduled on a most serious first basis. Another term for the process of determining who should be treated first based on the severity of the patient's condition is called triage. In this research the problem was formulated as follows:

- Multiple patients with different severity of medical condition (i.e. different patient classes) arrive at a healthcare department. At the healthcare department, a patient is assigned to a bed. The patient will occupy this bed, until the patient leaves the healthcare department. It should be noted that there is a fixed number of beds and thus it it might happen that a new arriving patient can not enter the healthcare department.
- While a patient is waiting for treatment, the patient's condition can deteriorate, which influences the treatment time, probability of successful treatment and the recovery time. A patient in a worse condition equals a longer required treatment time. If someone is in the worst shape, then he might die while waiting for treatment. It is also possible for a patient to die if the treatment was not successful.
- After treatment, a patient needs to recover, which can happen at the healthcare department, where the patient was treated but it can also happen at another healthcare department or at home.
- A scheduling policy will determine who will be treated next.

From the literature review, it could be concluded that some practical issues exist. Higher urgency patients are always prioritized over lower urgency patients, which aggravates the waiting time for lower urgency patients and might be the cause for the deterioration of a patient's medical condition. The practiced triage method is static, which means that a patient's condition is only identified at arrival, even though a patient's condition might change. The correction identification seemed to be a practical issue as well. Furthermore, triage was only based on the clinical needs of the waiting patients, while other factors like the patient mix in the pre-admission room can improve the efficiency of the system in terms of throughput and saving more lives. Several improvement methods exist, like fast track and

accumulating priority queue, but several authors highlight the importance of a state-based priority scheduling policy.

To tackle the scheduling problem, two different situations were analyzed. One situation with internal recovery (recovery in the healthcare department) and one situation without internal recovery (recovery outside the healthcare department). Both situations were formulated as a Markov Decision Process (MDP). Both MDP models included costs for a patient that has died or a patient that was blocked due to all beds being occupied. Both MDP models were first modeled as Continuous-Time Markov Decision Process (CTMDP) model and afterward uniformization was applied to convert the CTMDP into a Discrete-Time Markov Decision Process (DTMDP). Then both models were solved in MATLAB (ver. R2020a) with the use of policy iteration.

For the first model without internal recovery, different scenarios were analyzed: 1 server and 2 priority classes, 3 servers and 2 priority classes, and 1 server and 3 priority classes. A higher class patient indicates a patient in a more severe condition. An example of how a scheduling policy looks like for the 1 server and 2 priority class case can be seen in Figure 1. In the scheduling policy, the y-axis depicts the number of class 2 patients waiting for treatment and the x-axis depicts the number of patients of priority class 1 waiting for treatment. Every circle represents a possible state of the system where the server has to decide who to treat. The different colors are used to depict the action taken for each state, which can also be seen in the legend. Purple indicates to assign a class 1 patient for treatment, blue indicates to assign a class 2 patient for treatment and yellow indicates to not assign a a patient for treatment at all. The average cost value V is calculated by taking the average of all value functions from each possible starting state and can be found underneath the legend.

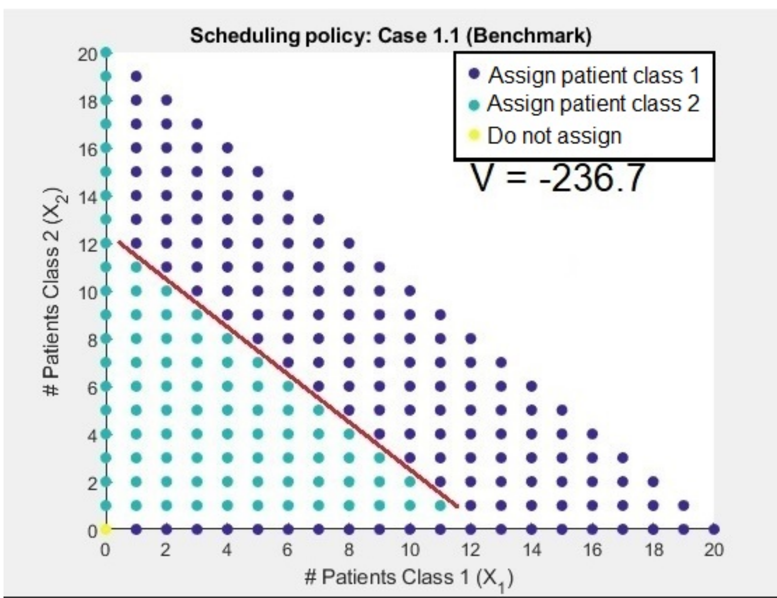


Figure 1: Scheduling policy: Case 1.1

The first thing that can be noticed is that the scheduling policy takes on the form of a switching curve, which is shown by a red line. The switching curve splits the optimal policy. On the left side of the switching curve, it is always optimal to treat class 2 patients (more severe patients) and on the right side of the switching curve, it is always optimal to treat class 1 patients. This makes sense as the closer we get to maximum capacity, the more preferable it becomes to treat less severe patients with a lower treatment time. It should also be noted that not assigning a patient for treatment is never the case when patients are waiting. Different parameter values (e.g. arrival rate) show different switching curves or no switching curve at all. The experiments with 3 servers and 2 priority classes followed the same depiction with a switching curve, dependent on the distribution of what kind of patients are currently in treatment. Furthermore, with multiple servers, the importance was shown of taking into account who is currently being treated when considering who to treat next. In one scenario, it was even optimal to not treat a patient and wait for a higher urgency patient to arrive. If an extra priority is introduced multiple switching curves can be seen between two classes of patients.

For the second MDP model with internal recovery, it was shown for the 2 class 1 server model, that it was important to take into account who is currently in recovery when scheduling patients. The importance of taking into account who is currently being treated changes depending on the relative difference between the treatment rate and recovery rate. The lower the recovery rate in comparison with the treatment rate, the more important it becomes to take into account those who are currently recovering. The performance of the optimal scheduling policy was tested against other simple policies. Depending on the arrival rate, the optimal MDP scheduling policy performs very well in comparison with the other simple policies. Especially with a high arrival rate, it was shown that always taking the most serious first (the one used in practice) was severely worse than the optimal state-based policy.

Some limitations exist concerning this research. A lot of simplifications were made, while in reality treating patients is a very complex system. Moreover, it was assumed that patients can only live or die, while in reality morbidity after treatment can also be considered. Using MDP means that due to the size of the state space, the research was limited by the processing power of the used notebook.

Interesting directions for future research are: Studying this problem with the use of a very powerful processor/computer to include more variables or more beds. Investigating this problem with different doctors or specialists, with each doctor or specialist influencing the treatment rate. Finally, as the form of the scheduling policy is known as a switching curve, it might be interesting to approach this problem in a different manner to find this switching curve.

Preface

This research thesis is the final step to completing my master Operations Management & Logistics at Eindhoven Univeristy of Technology. More than 6 years of studying have passed by at the moment of writing this preface. I can look back with a smile, as my student life has always been very exuberant thanks to my friends.

I would like to thank my first supervisor ir.dr. S.D.P. Flapper for coming up with this idea as a research project. As due to Covid-19 it was very hard to get a research project at a company, but it did result in the idea of this research project. The criticism and feedback on my work, which allowed me to improve my thesis and my research skills were also highly appreciated. I would also like to thank my second supervisor dr.ir. N. P. Dellaert for also giving his ideas when necessary.

Abbreviations

- **APQ** Accumulating Priority Queue
- **APQ-H** Accumulating Priority Queue with finite horizon
- **ATS** Australasian Triage Scale (triage system)
- **CTAS** Canadian Triage and Acuity Scale(triage system)
- **CTMDP** Continuous Time Markov Decision Process
- **DTMDP** Discrete Time Markov Decision Process
- **ED** Emergency Department
- **ER** Emergency Room
- **ESI** Emergency Severity Index (triage system)
- **FAHP** Fuzzy Analytic Hierarchy Process
- **FCFS** First Come First Served
- **FT** Fast Track
- **ICU** Intensive Care Unit
- **LOS** Length Of Stay
- **LSF** Least Serious First
- **LWBS** Left Without Being Seen
- **MAUT** Multi Attribute Utility Theory
- **MDP** Markov Decision Process
- **MSF** Most Serious First
- **MTS** Manchester Triage Scale (triage system)
- **OR** Operation Room
- **SATS** South African Triage Scale (triage system)
- **WT** Waiting time

Contents

1 Introduction

In 2020 the sudden up rise of the Covid-19 (corona disease) pandemic has put a lot of pressure on Intensive Care Units (ICU) worldwide. Several challenges arose with one of the challenges being how to deal with a surge of patients, while there is only a small amount of resources available to treat patients. Some examples of resources were ventilators, ICU staff, and ICU beds. It became clear that the demand for intensive care exceeded the capacity and decisions had to be made on who to treat first. Rationing of resources does not only happen during pandemics but also during huge accidents, war, and terrorist attacks. In addition, in the emergency department (ED), moments of exceeding capacity happen more frequently and overcrowding at the emergency department has been a widespread problem for several decades in multiple countries (Pines et al., 2011). According to a study in the United States among ED directors, 91% reported overcrowding as a problem, which has lead to patients waiting in hallways, all beds in ED being occupied and patients with a poor outcome due to overcrowding (Derlet, Richards, & Kravitz, 2001). During the Covid-19 pandemic, overcrowding became an even bigger problem. When a healthcare system with scarce resources is faced with a queue of patients, there are three options for improving the current situation: Increase the number of resources, regulate the queue by deciding which patient to accept into the queue or change the current policy of deciding which patient should be treated first. The first option can be very costly, as well as requiring a lot of time to realize, whereas the second and third option do not require any extra costs, except for a possible investment in a decision making system. This thesis focuses on the third option.

1.1 Outline

The outline of the remainder of this study is as follows. First of all, the problem statement, which comprises the problem context, problem definition, scope and assumptions, and research questions, is provided in Chapter 2. Then, a literature review is provided in Chapter 3, where the currently practiced triage and other priority scheduling policies are discussed. In addition, practical issues and the research gap is also provided in Chapter 3.3. In Chapter 4, the first Markov Decision Process(MDP) model without internal recovery is formulated and analyzed. Then in Chapter 5, the aforementioned MDP model is extended with internal recovery. In Chapter 6, a performance comparison between the optimal policy and other simple policies is provided with different chosen arrival rates. Finally, in Chapter 7, a conclusion with practical implications, limitations of this study, and possible directions for future research are given.

2 Problem statement

In this section the problem statement is defined. The first subsection starts with the problem context. The second subsection gives the problem definition. The third subsection describes the scope and assumptions related to the problem. The fourth subsection describes the research questions.

2.1 Problem context

In general, when an individual has a medical problem, (s)he will attend a healthcare service. Examples of healthcare services are the physiotherapist, general practitioner, dentist, or emergency department. For most healthcare services, which only treat non-emergency problems, an appointment system is in place to schedule patients. On the other hand for emergency departments, a priority queuing discipline is in place to determine who to treat first. A priority queuing discipline is also used at the IC or for ambulance transportation. Another term for the process of determining who should be treated first based on the severity of the patient's condition is called triage. According to Christian (2019), these triage types can be split up into primary, secondary, and tertiary triage, which can be seen in Table 1. It is not very unlikely for a casualty to be triaged multiple times in the process of being treated, as the casualty might be triaged for ambulance transportation and afterward for emergency department treatment.

Once a casualty is chosen for treatment, then that means that the required resources for treatment are unavailable for other casualties. For example, a doctor can only treat one patient at a time or one bed can only be used by one patient. Generally at the ED and IC, a more severe patient requires a longer treatment and recovery time. A longer treatment and recovery time means that it will take longer before resources become available for a new patient. Taking into account that a patient waiting for treatment might deteriorate over time, a problem exists between saving the most severe (with long treatment time) first or the least severe (with short treatment time).

Table 1: Classification of triage by location

(Christian, 2019)

Triage Type	Location	Priorities Addressed
Primary	Field	Who to immediately treat on scene (triage sieve) and priorities for evacuation from scene (triage sort)
Secondary	Entry to ER	Who to prioritize for resuscitation and disposition to treatment areas within the ER and/or admission to hospital ward
Tertiary	Exit from ER or entry to ICU/OR	Who to prioritize for definitive/critical care (OR and admission to ICU)

2.2 Problem definition

First, a visual example of the process is given in Figure 2. There are four stages in the process of treating a patient: Clinical analysis, waiting, treatment, and recovery. Figure 1 depicts how a patient and a bed move through the process. The character p depicts a patient and the character b depicts a bed, which can be seen on the arrows. For example, $p+b$ between treatment and recovery means that both the patient and bed move from treatment to recovery. Moreover, the event between brackets for p indicate the event that causes the patient to leave the process. The circle with patients indicates a place where patients come from and the circle with beds indicates the limited stock of beds. The rectangles indicate that a process is taking place. The triangle "Scheduling policy" indicates the decision of which patient (in bed) move to the treatment room.

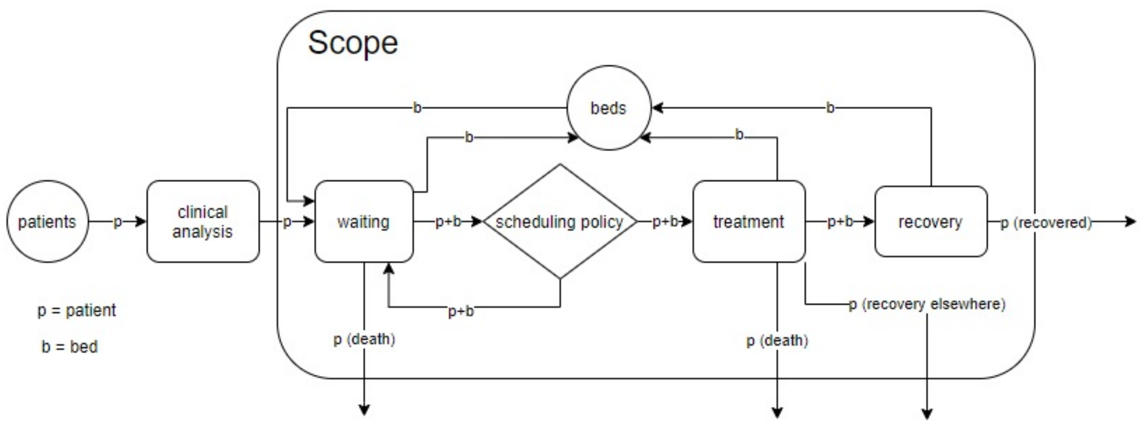


Figure 2: Process of treating a patient

The overall characteristics of the scheduling problem can be described as follows:

- Multiple patients with different severity of medical condition (i.e. the condition of the patient) arrive at a healthcare department. At the healthcare department, a patient is assigned to a bed. The patient will occupy this bed, until the patient leaves the healthcare department. It should be noted that there is a fixed number of beds and thus it it might happen that a new arriving patient can not enter the healthcare department.

- While a patient is waiting for treatment, the patient's condition can deteriorate, which influences the treatment time, probability of successful treatment and the recovery time. A patient in a worse condition equals a longer required treatment time. If someone is in the worst shape, then he might die while waiting for treatment. It is also possible for a patient to die if the treatment was not successful.

- After treatment, a patient needs to recover, which can happen at the healthcare department, where the patient was treated but it can also happen at another healthcare department or at home. A bed will become available, either after death, recovery or if a patient will recover somewhere else.

- A scheduling policy will determine who will be treated next.

2.3 Scope and assumptions

This part will discuss the scope of this study and the assumptions that are made. First, it should be noted that this research is not focused on one particular healthcare department, even though the process has the most similarities with an ED. For example, some similarities of this process can be found in ambulance transportation. During a large traffic accident with multiple casualties, some casualties have to wait at the accident site if there is only a limited amount of ambulances. Another example with some similarities can be found at the ICU. Patients might be waiting in another healthcare department (e.g. the ED) until there is a place available at the ICU.

Secondly, it is assumed that after clinical analysis the following variables of a patient are known: the severity of the medical condition, expected deterioration rate, expected treatment time, expected recovery time, and the probability of successful treatment. It is also assumed that during clinical analysis, patients that are deemed to be healthy or those that should be treated somewhere else are sent away and do not occupy a bed. This would otherwise result in extra unnecessary work. The underlying process of how the clinical analysis is done (e.g. measuring vital signs, blood pressure, and pain level) is left out of the scope.

Thirdly, the goal of the scheduling policy will be to save as many lives as possible. Different ethical prioritization factors exist for triage, that can be split up into medical factors and non-medical factors. Examples of medical factors are the rate of deterioration or survivability. While examples of non-medical factors are saving most lives, youngest first, saving the most QALY (Quality-adjusted-life-years), or lottery. Saving most lives is based on utilitarianism, producing the greatest good for the largest number of people. From the perspective of human rights, this is the fairest as everyone will be treated equally. Some researchers suggest that saving more years of life is more ethical and beneficial for society, but this will highly favor young people over old people and therefore goes against equality of all people. Quality-adjusted-life-years has a similar problem of inequality (Ghanbari et al., 2019). Moreover, from a model technical standpoint it is far more reasonable to assume that a patient in a worse state requires more treatment time, whereas someone who has less QALY does not necessarily equate less treatment time. Therefore, modeling the problem with the objective of minimizing the number of deaths is less complex than maximizing QALY, as for QALY it would be necessary to keep track of more variables (e.g. age and morbidity).

Lastly, the following assumptions are made:

- Patients do not leave the pre-admission room willingly.
- There is no classification error related to the severity of the medical condition and the condition is always known even after deterioration. In reality this classification error is somewhere between 20 to 25% (Saghafian, Hopp, Van Oyen, Desmond, & Kronick, 2012). However, big data and intelligent techniques (e.q. machine learning) are able to significantly improve the accuracy of clinical analysis and keep track of the patient's condition. (Fernandes et al., 2020).
- After successful treatment, a patient does not become sick/injured again.

- There is no pre-emption during treatment, which means that once a server starts treating a patient, the server will also finish treating this patient and does not switch in between.
- There is no pre-emption for occupying a bed. Once a bed is assigned to a patient, the patient will occupy the same bed until (s)he leaves the healthcare department due to death, being fully recovered or recovery elsewhere.
- All servers are capable of treating any patient equally well (i.e. all-round servers).
- Only patients in the worst condition can die while waiting for treatment.

2.4 Research questions

Derived from the problem description, this section will address the main research question and the corresponding sub-questions. The main question is as follows:

What is the most effective scheduling policy to treat deteriorating patients using a queuing model?

The most effective scheduling policy is determined by the number of dead and patients that are not able to enter the queue, due to all beds being occupied (i.e. blocked patients). The goal is to save as many patients as possible, while also trying to prevent capacity from reaching its limit resulting in patients being blocked (i.e. minimize the number of blocked and dead patients). To answer the main question, several sub-questions are necessary. The first task is to derive an optimal scheduling policy for a situation with with one server, two classes of patients, and no recovery at the healthcare department. This should be tested under different parameter values (e.g. different arrival rates) to show different optimal policies and understand the behavior of the scheduling policy. This leads to the first sub-question:

1. What is the most effective scheduling policy to treat deteriorating patients for situations with only one server and two classes of patients?

After finding optimal scheduling policies for the one server, two class patient and no recovery problem, it is interesting to see how the optimal scheduling policy looks like if more servers were introduced. This should be tested under different parameter values along the lines of sub question 1. It should be noted that a queue can only arise if all servers are occupied or a server decides to be idle. This leads to the second sub-question:

2. What is the most effective scheduling policy to treat deteriorating patients for situations with multiple servers and two classes of patients?

In the previous questions, only two priorities were assumed, which in reality may not be very practical as there is a lot of variety between patients. Therefore an analysis with one server and three priority classes will also be assessed and the scheduling policy's behavior under different parameter values. This leads to the third sub-question:

3. What is the most effective scheduling policy to treat deteriorating patients for situations with one server and three classes of patients?

All previous questions assumed no recovery at the healthcare department, but somewhere else. This indicated that a bed would immediately become available again after successful treatment. In the following question, we want to study the behavior of the scheduling policy, if we include recovery in the healthcare department. When recovery is included in the healthcare department, a bed becomes available after successful treatment and recovery or death. This leads us to the fourth sub-question:

4. What is the most effective scheduling policy to treat deteriorating patients for situations that include recovery inside the healthcare department?

For the final sub-question, we want to asses the performance of the optimal scheduling policy against other policies. An example of a policy, which is used in practice a lot, is helping the most severe at all times. The fifth and final sub-question is:

5. How does the optimal scheduling policy perform against other policies?

3 Literature review

Healthcare systems are very complex systems with very large variability. This can be seen in the large number of different research papers focused specifically on either one disease or one particular hospital. There are also some research papers on a more abstract level that make use of queuing models for modeling healthcare systems. See for example the literature review on queuing models in healthcare related problems of Lakshimi et al. (2013). A queuing model can accommodate key features of a healthcare system, from which its result might not be directly use able in practice, but might provide insights for any possible real world change (Foster, Hosking, & Ziya, 2010).

The following data evaluation is split up into two different sections. The first subsection describes some of the current scheduling triage systems used in practice. The second subsection describes relevant literature on patient scheduling from a queuing theory perspective. The data evaluation is accompanied by a table depicting the most important characteristics of the most relevant articles in Appendix A.

3.1 Triage in practice

Different policy systems exist for scheduling non-elective patients. Non-elective patients are patients that need a medical procedure, which can not be scheduled in advance, due to the urgency of their medical condition. In some rural and underdeveloped areas, all non-elective patients are seen on a first-come-first-served basis or intuition (Aloyce, Leshabari, & Brysiewicz, 2014). However, most scheduling systems for non-elective patients in more developed areas are based on a five-level classification system combined with a first-come-first-served system. Patients are classified from high to low priority (i.e. 1 to 5) with the highest priority indicating the highest urgency of medical need. The highest priority patient in the pre-admission room is seen first. If multiple patients from the same priority are waiting, then the first patient who came in from that particular priority is treated first. Examples of these scheduling systems that are widely used are the Australasian Triage Scale (ATS)(Considine, LeVasseur, & Villanueva, 2004), Canadian Triage and Acuity Scale (CTAS)(Beveridge et al., 1998) and Emergency Severity Index (ESI)(Wuerz, Milne, Eitel, Travers, & Gilboy, 2000). Even though they use the same scheduling system, some differences can be found in the variables each triage system uses to classify a patient. For example, all three aforementioned triage systems incorporate vital signs as a variable for classifying the priority of a patient. The Emergency Severity index also includes the number of resources necessary for treatment as a variable, while the other triage systems do not. On the opposite, the Emergency Severity Index does not include the recommended time for safely waiting, while the others do.

A systematic review on the performance of identifying high and low urgency patients using the ESI, CTAS, and MTS (Manchester Triage Scale) showed that the performance of identifying patients correctly was highly variable (Zachariasse et al., 2019). This variability was not only found between triage systems, but also between studies examining the same triage system. Zacharissasse et al. concluded that of all three triage systems, no triage system was superior. Hinson et al. (2019) came to the same conclusion that no triage system should be preferred over the other in their systematic review of triage systems evaluating the performance in terms of severe outcomes (e.g. mortality or critical illness outcomes). Besides studies on ESI, CTAS, and MTS, they also included studies on the ATS and SATS (South African Triage System). They stated that similar weaknesses can be found among triage systems with the most alarming weakness being the high sensitivity of initial triage, suggesting that the initial triage is not in line with the actual severity of the patient's condition at the time of treatment. This might be since either the initial analysis was wrong or the patient's condition deteriorated while waiting. A simple improvement would be to re-assess all waiting patients now and then to reconsider their urgency (Reinhardt, 2017). Hinson et al.(2019) suggested improving triage systems with electronic health records and decision support systems. Both systems are very important for logistical decision making and process streamlining of medical procedures in the future (Fitzgerald & Wu, 2017).

In the aforementioned triage systems, the patients with the most serious injury are always treated first. A study of Cao (2012) showed that the most serious first principle displays a much poorer outcome in terms of the death toll, when medical resources are very scarce, in contrast to the least serious first. Less urgent patients gradually deteriorate and turn into more urgent patients while waiting for a medical resource to become available. Therefore, favoring the least severe over the most severe might save more lives during periods of extreme scarcity of resources. Consequently, he also noted the importance of researching the behavior of different scheduling policies researched under different circumstances.

Only in some rare situations, the triage protocol changes from highest urgency to highest chance of survival. One example is that during war, the least injured are treated first so they can recover and assist others (Pollaris & Sabbe, 2016). Another example is that during the Covid-2019 pandemic, a triage script was given by the Dutch government, which stated that the least injured should be treated first during a very extreme scarcity of resources. Christian (2019) mentioned that the research literature on triage in disaster situations is very limited and that the most common error that can be found in triage literature is the attempt to derive triage protocols from non-resource-scarce scenarios to eventually use in mass casualty situations (Christian, 2019).

The following part is not based on queuing theory or used in practice, but is still interesting and might be useful for the actual research thesis. Another stream of literature based on tackling the aforementioned problems is based on multi-criteria decision making. Ashour and Okudan (2010) propose the use of a fuzzy analytic hierarchy process (FAHP) and multi-attribute utility theory (MAUT) to sort patients. Instead of patients being assigned a priority class based on urgency, patients are given a utility value based on vital signs and descriptive variables. The weight of each vital sign depends on the main complaint the patient has. The patient with the highest utility value is seen first. Ashour and Okudan (2016) tested the FAHP and MAUT method against ESI in an ED with a discrete event simulation. There was no strong statistical evidence that one system would do better than the other in terms of throughput, waiting time, flow time, and the average length of stay. However, FAHP MAUT balances the length of stay and the waiting time for all ESI urgency levels and outperforms the ESI system for patients levels 4 and 5 (the lowest urgency patients). Rahimi et al (2016) also proposed the use of a fuzzy multi-criteria decision making technique to determine the weights of different criteria and risks. Afterward, they calculated the patient's score to determine the patient's place in the waiting queue. To deal with the deterioration of patients they introduced a delay ratio, which increases the priority of the patient. All criteria used in the decision making were only based on clinical and social needs.

It can be concluded that the triage process can basically be split up into two parts. The first part of the process is the correct identification of the severity of the patient's condition and updating the classification if the patient's condition changes, which is mostly based on the clinical needs of a patient. The second part of the process is scheduling all patients that are waiting for treatment, which is a logistical problem. Some healthcare-related literature exists on scheduling patients waiting for treatment, but there are also several relevant articles based on priority scheduling in other research fields.

3.2 Priority scheduling policies

Xie, He and Zhao (2008) assumed that customers waiting for service can change priority class after an exponential time with a predetermined parameter. They investigated the stability of a priority queuing system and conclude that the system stability is determined by arrival rates of all customers and the service rate of the highest priority class of customers. To ensure the stability of the queuing system, resources should be allocated to serving customers of the highest priority, but this may result in long waiting time for lower priority and thus more priority changes (He, Xie, & Zhao, 2012). Therefore, finding the right balance between system stability and keeping all queues short is key in a queuing system with customer upgrades.

3.2.1 Fast track

One of the first priority policies applied in the healthcare setting is the shortest processing time rule by McQuarrie (1983). Patients that require the least service time are given the highest priority. It is shown to minimize waiting time, during moments of high congestion, but it is rarely used in the hospital setting due to its perceived unfairness. In addition to patients that would come in with a more severe condition would have to wait longer with all its consequences. A strategy born from this rule is the fast track strategy. In the fast track strategy, non-emergency patients with an uncomplicated disease (i.e. expected low service time) are bundled up in their own queue with a dedicated server. Fast track is a form of server capacity partitioning, where a part of the server capacity (e.g. rooms or staff) is dedicated to only serving lower priority patients. This has been shown to reduce the average waiting time per patient (Meislin, Coates, Cyr, & Valenzuela, 1988). Furthermore, FT was also associated with a decrease in the average length of stay (LOS) and percentage of patients leaving without being seen(LWBS), without a harmful effect on the quality care, measured in terms of revisit rate and mortality rate (Sanchez, Smally, Grant, & Jacobs, 2006). Sanchez et al. concluded that the average waiting time(WT) was reduced. Moreover, another study on the implementation of FT also showed a reduction in average WT and LOS for discharged patients without increasing waiting times for admitted patients, even in an ED with 22% of all patients being low acuity (i.e. priority 3,4 and 5 on ATAS) (O'Brien, Williams, Blondell, & Jelinek, 2006). However, it should be noted that both studies only conducted their research in one (but not the same) hospital, where both hospitals were undergoing construction work. Therefore it was not clear, whether the decrease in LOS, WT, and LWBS was only due to the implementation of FT or other variables as well (e.g.

additional servers or treatment rooms). Assessing the impact solely of FT in practice is difficult, due to a constantly changing environment in terms of variable patient flow and in their cases a varying amount of resources.

3.2.2 Accumulating priority queue

Stanford et al. (2014) investigated the use of the accumulating priority queue (APQ) in the healthcare setting. The accumulating priority queue is based on the delay-dependent queue discipline of Kleinrock (1964). In the APQ different classes of customers/patients accumulate priority over time. The higher the initial urgency of the patient, the higher the rate at which the patient accumulates priority. The longer a patient has to wait, the higher his priority will be. They derived a feasible range of values for the accumulation rate for the case with one single server and two priority classes of patients, respectively priority class 4 and 5 of CTAS with identical mean service times and arrival rates. The recommended maximum waiting times for priority classes 4 and 5 are respectively 60 and 120 minutes according to the CTAS. The feasible range of values was produced to have a certain percentage of waiting patients be treated before the recommended maximum waiting time.

Sharif et al. (2014) extended the APQ model by introducing an extra server and an extra priority class. They also established the waiting time distributions for different priority classes of customers/patients in a multi-server system with common exponential service time and Poisson arrivals. Li and Stanford (2016) also investigated the APQ multi-server system Poisson arrivals, but with heterogeneous servers (i.e. different service time per server). They formulated a "r-dispatch policy", to decide which server should start treating a new patient when there are multiple different servers idle.

Cildoz, Ibarra and Milor (2019) extended the previously mentioned APQ model even further with a finite horizon, which is called APQ-h. Patients accumulate extra priority until the maximum allowed waiting time per class is reached. They investigated the use of APQ and APQ-H in a simulation model replicating a real ED and find the optimal APQ type policies through simulation-based optimization. They compared both APQ and APQ-h with other pure priority policies in their simulation model and show that both the APQ and APQ-h model performed better during moments of high congestion, but they did not show a significant difference between APQ and APQ-h.

Ferrand et al (2018) compared the effect of FT with strict and partial flexible capacity, dynamic priority queue (which is almost similar to APQ) and the practiced triage scheduling used at the ED on the LOS. Using an empirical simulation, they concluded that the dynamic priority queue dominates the fast track and static priority rules in terms of balancing wait time across priority levels. Moreover, they show that for different ED sizes, FT does not reduce the average LOS of low-priority patients without increasing the average LOS of high-priority patients, unless the mean and variance of the treatment time was reduced.

3.2.3 Mixed Integer Programming (MIP)

Harzi et al. (2017) presented a multi-server multi-stage priority queue system to minimize the expected total waiting time per patient. The problem is modeled with an non-homogeneous Poisson arrival process of patients with different levels of priority, based on the severity of the patient's condition. Depending on the level of priority, a patient follows a trajectory of different healthcare stages. They presented a mixed integer linear programming (MILP) approach to tackle their problem to minimize the total waiting time per patient in the ED. The MILP was then solved with IBM Ilog Cplex 12.7 for small instances (25), as larger instances resulted in more difficulties. Interesting to note here is that they consider both material and human resources at the same time.

Azadeh et al. (2014) focused on scheduling patients in ED laboratories. They also considered multiple stages a patient has to go through with the pathway being dependent on the medical condition of the patient. Each laboratory is considered as a stage, each patient as a job and each server at a laboratory as a machine with a limited number of parallel identical machines per laboratory. The patient scheduling problem was formulated as a generalized flexible open shop problem and a MILP model was proposed. Due to the scheduling problem being NP-hard, a genetic algorithm was developed for solving the problem.

3.2.4 $c\mu$ rules

One of the oldest priority policy applied in job selection is the $c\mu$ rule by Smiths (1956). In this case, c depicted the holding cost per job. When the cumulative waiting cost is a linear function of the waiting cost then the queue with the highest "cost multiplied with service rate" is chosen as the highest priority queue for service. Van Mieghem (1995) extended the $c\mu$ rule for non-decrease convex delay costs by taking the waiting time into account. This was called the generalized $c\mu$ rule and was shown to be optimal if the system was operating near capacity.

When abandonment is introduced the $c\mu$ rule does not hold as an optimal rule in general, only under certain conditions (Down, Koole, & Lewis, 2011). Down, Koole and Lewis investigated the dynamic server control in a two-class service system with abandonments by formulating the system as a continuous-time Markov decision process with an unbounded transition rate. Their numerical results suggest that if a decision-maker would ignore the abandonments, then this could seriously decrease earned rewards or increase accrued costs. The goal of the decision-maker is to maximize rewards while minimizing the server idleness.

Atar, Giat and Shimkin (2010) extended the aforementioned $c\mu$ rule into the $c\mu/\theta$ rule with θ being the exponentially distributed abandonment time. Abandonment or jilting means that customers waiting for service might leave the queue before being served. Their analysis shows that the $c\mu/\theta$ rule minimizes the long term average costs in a multi-class many-server queuing system.

3.2.5 State dependent priority policy

Argon, Ziya and Righter (2008) looked into scheduling patients during mass casualty events that might abandon the system if they are not serviced within their "lifetime". They investigated the problem with two types of jobs that have an exponential distributed service rate and lifetime rate (i.e. abandonment rate). They showed that the job with the shortest lifetime and shortest service time should always have the highest priority, which intuitively makes sense. The more interesting scenario is when the service time for job 1 is longer than job 2 and the lifetime of job 1 is shorter than job 2. For that scenario, the authors provided two state-dependent heuristics, so-called triangular and rectangular heuristics and two non-state dependent heuristics. They showed that when abandonment rates are small, then both state-dependent and non-state dependent performed reasonably well. On the other hand, when abandonment rates are relatively high compared to the service rate, then the state-dependent policies performed much better.

Xie et al. (2016) investigated the conditions for treating a less severe patient over a more severe patient in emergency medical services using queuing theory. They model their problem as a Continuous-Time Markov Decision Process with two priority classes of patients that can both deteriorate. However, instead of both priority patients leaving the system, the low priority patient can change into a high priority patient and the high priority patient can abandon the system (healthcare department), both after a random amount of time that is exponentially distributed. Therefore, this research is a combination of abandonment and customer upgrades (i.e. transfer). Under the assumption of holding costs, transfer costs and abandonment, they identify conditions under which it is better to treat the less severe patient over the more severe patient. It can be concluded from their research that a dynamic control policy according to the system state should be used rather than any static priority policies.

3.3 Conclusion

Most literature about patient scheduling in the healthcare setting is investigated in the ED and not the IC. This is probably since overcrowding happens more frequently at the ED than the IC under normal (not a pandemic or disaster) conditions. Moreover, the number of articles that include scheduling with disaster conditions or that are focused on saving as many lives as possible is very scarce. Most articles focus on minimizing waiting time, probably due to the low mortality rate of patients in the ED (Zachariasse et al., 2019). However, articles that focus on minimizing waiting time, can still be very useful with the goal of saving as many lives as possible due to the possibility of deterioration while waiting. Several different policies were found:

- **Fast track**: Server capacity partitioning, where the lowest priority patients get their own queue
- **Accumulating priority queue**: Time-dependent dynamic priority allocation
- **LSF OR MSF**: Static policy, which always treats respectively the least or most serious first
- $c\mu$ **rule and derivatives of** $c\mu$ **rule**: Static policies with c as holding cost per patient
- **State dependent priority policies**: Dynamic policies that are dependent on the number of patient per class that are waiting in the queue

3.3.1 Practical issues

The currently practiced triage systems seem to do quite a reasonable job at scheduling patients. However, a lot of room for improvement exists, especially in the case of very dire situations.

- The first problem of the currently practiced triage is that higher urgency patients are almost always prioritized over lower urgency patients. This is perceived the fairest, but this leads to lower urgency patients having to wait for a very long time. Moreover, if we assume that patients with a higher urgency need more treatment time and have a lower chance of successful treatment, then this aggravates the long waiting times of lower urgency patients and it slows down the rate of treated patients per time period. Furthermore, prioritizing the most severe and thus having a slower rate of treated patients per time period, lead to capacity reaching its limit at a faster rate. During Corona, this resulted in some ICUs not having enough ICU beds available for patients.

- The second problem to address is that the currently practiced triage is a static method. During triage, a patient's condition is being assessed at one point in time and then given his urgency category, although the patient's condition might deteriorate over time. Patients that deteriorate while waiting can lead to longer treatment times and less chance of a successful treatment for these particular patients. A simple solution would be to triage patients several times during waiting or to make use of electronic health records to re-assess their categories. If one of the patient's condition has changed then the treatment order should be updated accordingly.

- The third problem that can be mentioned is the fact that triage is only based on the clinical needs of the waiting patients. Factors that could be taken into account as the service (treatment) time and the patient mix of patients waiting and being treated could greatly improve the efficiency of the system. For example, if a lot of non-severe patients are being treated, then it is much more likely for a server to become free in the next time period in comparison with a situation where all servers are occupied with very severe patients.

- The fourth problem is the accurate identification of the urgency of a patient's condition. Currently, each triage system uses some similar variables and some different variables, but none of the triage systems seems to be superior over the other ones. In the future, big data and more knowledge in the healthcare domain can lead to an improvement in accurate identification of the severity of a patient's condition (Bates, Saria, Ohno-Machado, Shah, & Escobar, 2014).

3.3.2 Research gap

Overall, there is little research on how and when a triage protocol should change depending on the number of patients waiting for treatment (i.e. the state of the system), while multiple authors highlight the importance of it for future research. The paper of Xie et al. (2016) concluded that a dynamic control policy according to the system state should be used rather than a static control policy. Ferrand et al. (Ferrand et al., 2018) also mention that dynamic properties of the accumulating priority queue could be improved by also taking into account the system state and as has been mentioned by Cao (2012), the behavior of different scheduling policies should be researched under different circumstances. A high congested waiting queue should be scheduled differently than a low congested waiting queue and therefore taking into account the system state is very important during pandemics or disasters. Besides, no research paper took into account the patients currently in treatment or recovery while deciding who to treat first.

To conclude this literature review, no research paper would fit all of the most important characteristics mentioned in Appendix A. The paper of Xie et al. (2016) comes very close, as their paper considered patient's condition-dependent priority allocation, while also taking into account deterioration and abandonment. However, they did not take a look at multiple servers, nor at condition-dependent result treatment. Their problem was also modeled with an infinite queue, while we are interested in the patient scheduling problem with a finite queue. Furthermore, some aspects (e.g. cost allocation) are different in their model to what we want to investigate. Hence, to our knowledge, the patient priority scheduling problem with abandonment and deterioration in a finite queue with multiple servers has never been researched before.

4 Model 1: Treating patients without internal recovery

The first model "Model 1" for treating patients without internal recovery will be discussed in this chapter. In model 1, the patient can immediately leave his bed after treatment either to recover at home or recover somewhere else (e.g. family or another hospital department). The first subsection gives an extensive description of the situation that has been analyzed. The second subsection provides the mathematical formulation of the Markov Decision Process (MDP) model. The third subsection describes the computational methodology and the final subsection describes the results of the first MDP model.

4.1 Description

4.1.1 Process of treating a patient without internal recovery

The process of treating a patient without internal recovery can be described in three stages: Patient arrivals, the pre-admission room and treatment. Figure 3 shows a schematic representation of the process of treating two condition based patient classes without internal recovery as an example.

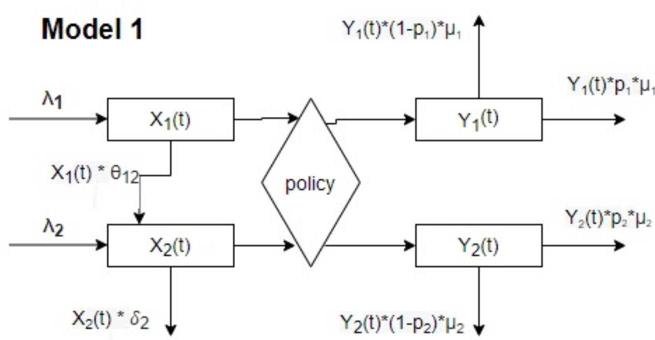


Figure 3: Schematic representation of MDP model 1 with two priority classes and one server

Patient arrivals
Condition-based patient arrivals are modeled according to a Poisson process λ_k with k indicating the patient's condition. If a patient arrives at the healthcare department, a bed will be assigned to the patient and the patient has to wait in a pre-admission room for treatment. If all beds are occupied (i.e. no bed available), then a newly arrived patient will be blocked from the healthcare department. A condition-based penalty c_{bk} is given for a blocked patient.

Pre-admission room
The pre-admission room is split up into multiple condition-based queues. The patient will join a queue depending on the patient's condition. $X_k(t)$ depicts the number of patients in condition k waiting for treatment at time t with $\{k \in \mathbb{Z} : 1 \leq k \leq k_{max}\}$ and k_{max} being the worst condition. A policy decides, which condition-based patient queue should be served next. If a certain condition based queue is chosen for treatment, then the patient that has waited the longest in that particular condition based queue will be treated first (i.e. first

in line). While waiting in the pre-admission room, any patient that is not in the worst possible condition can deteriorate according to a condition based exponential distributed deterioration rate θ_{kk+1}. If a patient deteriorates, then the patient is assigned to a worse condition-based queue (i.e. $k+1$). The total transfer rate of patients deteriorating from one queue to another queue is equal to $X_k(t) * \theta_{kk+1}$. If the patient is in the worst condition k_{max}, then the patient might die according to an exponential distributed mortality rate $\delta_{k_{max}}$. If the patient dies, the patient will immediately leave the pre-admission room and the bed becomes available again. The total transfer rate of patients leaving the worst condition based queue due to death is equal to $_{k_{max}}(t) * \delta_{k_{max}}$. A penalty cost c_d is given for the death of a patient.

Treatment

A policy will determine, which patient will be treated next. The patient's treatment time μ_k is modeled with an exponential distribution based on the condition of the patient at the start of the treatment. $Y_k(t)$ depicts the number of patients in condition k in treatment at time t. The total number of patients that can be treated simultaneously depends on the number of servers M. Each patient requires one server for treatment. Once the treatment is finished, a probability p_k based on the condition of the patient at the start of the treatment will decide whether the treatment was a success or not. If the treatment failed with probability $1 - p_k$, then it is assumed the patient has died and a penalty cost c_d is given. If the treatment was a success, then the patient will leave the treatment room. The total rate of patients leaving the treatment room after successful treatment is equal to $Y_k(t) * p_k * \mu_k$ and the total rate of patients leaving the treatment room after unsuccessful treatment is equal to $Y_k(t) * (1 - p_k) * \mu_k$. Either after failure or success, a bed will become available again for the next waiting patient.

4.1.2 Variables list

Table 2 depicts all aforementioned variables and variables that will be used for the MDP model. All of the patient condition based rates are exponentially distributed.

Table 2: Variables list

Variable	**Description**
B	Total number of beds available at the hospital department
c_d	Penalty cost of patient death
c_{bk}	Penalty cost of blocked patient in condition k
K	Set of all possible conditions k a patient can be in
k	The condition (class) assigned to a patient ranging from $\{k \in \mathbb{Z} : 1 \leq k \leq k_{max}\}$
k_{max}	Worst condition (class) a patient can be in
M	Total number of servers
p_k	Probability of successful treatment of each patient in condition k

S	The state space of the healthcare department that contains all possible states s
$s(t)$	The state of the healthcare department at time t
$X_k(t)$	Number of patients in condition k waiting for treatment at time t
$Y_k(t)$	Number of patients in condition k being treated at time t
γ	Discount factor with $0 \leq \gamma \leq 1$
$\delta_{k_{max}}$	Mortality rate of a patient in the worst condition k_{max} waiting for treatment
η	Uniformization rate
θ_{kk+1}	Deterioration rate of each patient in condition k waiting for treatment
λ_k	Independent Poison process arrival rate of each patient in condition k
μ_k	Treatment rate of each patient in condition k

4.2 MDP Formulation

To answer our research questions, the patient scheduling problem without internal recovery will be analyzed as a Markov Decision Process (MDP) model. MDP models are used for modeling decision making in situations where outcomes can be random but are still partly under the control of the decision maker. In the patient scheduling problem, a decision can be made on who to treat, but it is never sure how this will affect the future, as randomness exists in new patient arrivals, patients that might deteriorate or when a treatment is finished. The MDP model is first modeled as a Continuous Time Markov Decision Process (CTMDP) and afterward uniformization is used to transform the CTMDP into a Discrete Markov Decision Process (DTMDP).

4.2.1 State Space

The state space is finite and dependent on the total number of beds B, the total number of servers M and the number of priority classes K. The state of the system at time t can be described by the number of patients per class waiting for treatment and currently in treatment. Let $X_k(t)$ denote the number of patients waiting for treatment and $Y_k(t)$ denote the number of patients in treatment per class k at time t, then the state at time t can be described as in Equation (4.2.1.1). Furthermore, all possible states are constrained by the number of beds B and the number of servers M (see Equation (4.2.1.2)).

$$s(t) = (X_1(t), X_2(t), X_k(t), ..., X_{k_{max}}(t); Y_1(t), Y_2(t), Y_k(t), ..., Y_{k_{max}}(t)) \tag{4.2.1.1}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{k=1}^{k_{max}} X_k(t) + Y_k(t) \leq B \\ & \sum_{k=1}^{k_{max}} Y_k(t) \leq M \end{aligned} \tag{4.2.1.2}$$

The state space S consists of all possible distributions of beds over the priority classes and the number of working servers. The following formula shows how an increase in the number of priority classes, number of beds or number of servers influences the size of the state space S. Let B denote the total number of beds, K the number of priorities and M the total number of servers, then the total number of states can be calculated as in Equation (4.2.1.3). The first part of the equation before the plus sign depicts the distribution of beds when the number of beds is equal or lower than the number of servers, while the second part of the equations depicts the distribution of beds when the number of beds is higher than the total number of servers. For example, if there are 2 priorities, 1 server and 3 beds then the number of possible states is equal to:
$\sum_{b=1}^{1} \frac{(1+4-1)!}{1!(4-1)} + \sum_{c=2}^{3} (\frac{(c+2-1)!}{c!(2-1)} * \sum_{d=1}^{1} \frac{(1+2-1)!}{1!(2-1)}) = 8 + 3 * 2 + 4 * 2 = 22$

$$\sum_{b=1}^{M} \frac{(b+2k_{max}-1)!}{b!(2k_{max}-1)} + \sum_{c=M+1}^{B} (\frac{(c+k_{max}-1)!}{c!(k_{max}-1)} * \sum_{d=1}^{M} \frac{(d+k_{max}-1)!}{d!(k_{max}-1)}) \tag{4.2.1.3}$$

4.2.2 Action Space

The action space is finite and dependent on the state space. All actions are decided by a scheduler and not the server itself. The action space for the scheduler consists of do not assign or deciding to assign a class k patient to a server (i.e. person or machine that performs treatment), but is dependent on the current state of the system. All servers are capable of treating all patients (i.e. all round servers). If there are multiple patients with different conditions waiting, then the scheduler can decide from which condition based queue a patient should be treated or to not assign at all. If no patients are waiting or all servers are occupied then the only available action is to not assign. Once a patient is chosen for treatment, the patient moves from the pre-admission room to the treatment room. It is assumed that no time elapses for this decision to be made. Mathematically this can be formulated as follows in Equation (4.2.2.1).

$$A(t) = \begin{cases} do\ not\ assign & if\ X_k(t) = 0\ \forall\ k \in K \lor \sum_{k=1}^{k_{max}} Y_k(t) = M \\ \{do\ not\ assign,\ assign\ patient\ class\ k\} & if\ X_k(t) > 0\ \exists\ k \in K \land \sum_{k=1}^{k_{max}} Y_k(t) < M \end{cases} \tag{4.2.2.1}$$

4.2.3 Cost

A policy π specifies when to perform each of the aforementioned actions. Penalty costs are incurred when a patient is blocked (i.e. the patient arrives, but there is no bed available). Penalty costs are also incurred when a patient dies either after treatment or during waiting. Costs are only incurred when an event happens, therefore the discounted expected costs over an infinite horizon with policy π, initial state s(0), and discount factor γ $(0 \leq \gamma \leq 1)$ is given by Equation (4.2.3.1) :

$$v_{\gamma}^{\pi}(s(0)) = E^{\pi}[\sum_{i=1}^{\infty} (\sum_{k=1}^{kmax} \gamma^{\phi_b(i)} c_{b_k} + \gamma^{\phi_d(i)} c_{d_k}) | s(0), \pi] \tag{4.2.3.1}$$

where $\phi_b(i)$ represents the i^{th} event time a patient is blocked and $\phi_d(i)$ represents the i^{th} event time a patient died. The goal is to find a policy π that minimizes the discounted expected costs over an infinite horizon:

$$v^{\pi*}(s(0)) = \min_{\pi} v^{\pi}(s(0))$$

4.2.4 Uniformization from CTMDP to DTMDP

To transform the CTMDP into a DTMDP, the uniformization method will be used (Lippman, 1975). The uniformization method works by scaling all outgoing transition rates to the same rate. To do this, we need to choose a uniformization rate η that is equal or higher than the highest total outgoing transition rate. In other words, if q_{ij} is an outgoing transition rate from state i to j and q_i the sum of all outgoing rates of state i, then the following inequality holds:

$$\eta \geq q_i \ \forall i$$

It should be noted that the deterioration rate and mortality rate are dependent on the patient's condition. This is because if the number of patients waiting for treatment for a particular class increases then it is more likely that a patient will deteriorate to a higher class or in the case of the highest priority class more likely that a patient will die. Due to a limited number of beds and therefore a limited state space it is possible to calculate the highest q_i and use that value as the uniformization rate η. In this MDP, the highest possible transition rate for model 1 is given in Equation (4.2.4.1) constrained by Equation (4.2.1.2). At any time q_i can be calculated by adding up the following elements per taken action:

- Sum of all condition-based patient arrival rates
- Sum of all condition-based patient treatment rates multiplied by the number of patients in treatment for that particular condition
- Sum of all condition-based patient deterioration rates, except worst patient condition, multiplied by the number of patients waiting for treatment for that particular condition
- Mortality rate multiplied by the number of patients waiting for treatment in the worst condition

$$\eta = \max_i q_i = \max_{Xk, Yk \, \forall k \in K} [\sum_{k=1}^{kmax} (\lambda_k + \mu_k * Y_k) + \sum_{k=1}^{kmax-1} (X_k * \theta_{kk+1}) + X_{kmax} * \delta_{kmax})]. \quad (4.2.4.1)$$

$$\text{s.t.} \quad \sum_{k=1}^{k_{max}} X_k(t) + Y_k(t) \leq B$$
$$\sum_{k=1}^{k_{max}} Y_k(t) \leq M \quad (4.2.1.2)$$

To solve a CTMDP, a transition matrix is necessary. The transition matrix consists of all outgoing rates from one state to another for all possible states. Therefore, by taking the sum of all outgoing transition rates per state (i.e. per row of the transition matrix) all q_i are calculated and the maximum can be found of all q_i. This maximum can then be chosen as the uniformization rate η.

Each transition in state i, leaves the state at a total outgoing rate of q_i. However, because we want all total outgoing rates to be equal to η, we suppose that there is a fictitious rate $\eta - q_i$ at which a state i goes back to state i. In this MDP model an exponentially distributed amount of time with mean $\frac{1}{\eta}$ is spent in state i and makes transitions with probabilities that are controlled by an embedded DTMC (Ibe, 2013). The transition probabilities p_{ij} are as given in Equation (4.2.4.2):

$$p_{ij} = \begin{cases} 1 - \frac{q_i}{\eta} & i = j \\ \frac{q_i}{\eta} q_{ij} & i \neq j \end{cases} \tag{4.2.4.2}$$

4.2.5 Transition probabilities

The transition probabilities of model 1 are based on the following events: Arrival of a patient, successful treatment of a patient, failed treatment (death) of a patient, deterioration of a patient and death of a waiting patient. All events are stochastic elements of the model. It is assumed that it takes no time to bring a patient from the pre-admission room to the treatment room. Table 3 represents an overview of all possible transitions due to events or action taken. The first column depicts the descriptions of all possible events or actions. The second column depicts the transition probabilities, which are based on Equation (4.2.4.2). Furthermore, if s(t) is the current system state as depicted in equation 1 and s(t+1) is the next system state then $s(t) \to s(t+1)$ depicts the state change as indicated in the third column. The state change is one patient movement at a time. Assign patient state k is only possible when the server is free and patients are waiting for treatment (See 4.2.2 Action Space).

Table 3: State changes

Events	$\boldsymbol{Pr(s_{t+1}\|s_t)}$	$\boldsymbol{s(t) \to s(t+1)}$
Arrival of patient k	$\frac{\lambda_k}{\eta}$	$X_k + 1$
Successful treatment patient k	$\frac{\mu_k * p_k}{\eta}$	$Y_k - 1$
Failed treatment patient k	$\frac{\mu_k * (1-p_k)}{\eta}$	$Y_k - 1$
Deteriorate patient k	$\frac{\theta_{kk+1} * X_k}{\eta}$	$X_k - 1, X_{k+1} + 1$
Death patient k_{max}	$\frac{\delta_{kmax} * X_{kmax}}{\eta}$	$X_{k_{max}} - 1$
Actions		
Assign patient state k	-	$X_k - 1, Y_k + 1$

4.2.6 Solving methods

There are several solution methods for solving MDP's, but the two methods that will be tried in this thesis are value iteration (Bellman, 1957) and policy iteration (Howard, 1960). Both methods have been used extensively in other research and there are a lot of sources on how to apply both methods.

1. In value iteration a value function is randomly selected and from there on a new improved value function is found in an iterative process until reaching the optimal value functions. An optimal policy can then be derived from the optimal value function by choosing the best action for every state.

2. In policy iteration, a policy is randomly selected and the value function corresponding to it is calculated. From there on a new improved policy is found based on the previous value function and this is repeated in an iterative process until the optimal policy is reached.

Both methods should lead to the optimal policy, but there can be a huge difference in calculation time and required processing power. Therefore depending on the calculation time, one of two methods will be chosen. It is expected that the policy iteration is faster, due to the small number of actions available and thus a small and finite number of possible action combinations (i.e. policies), which results in the policy function converging a lot faster than the value function.

4.3 Computational experiments

4.3.1 Computational methodology

To find the optimal policy for the DTMDP, we will use Matlab (ver. R2020a) and a few functions of the Markov decision process toolbox (Marie-Josee Cros, 2020). [1] The MDP process toolbox contains a value iteration and policy iteration solver. It also contains a policy evaluation function. Before using these functions, the function solvers are thoroughly checked and other MDP problems were investigated to assess whether the functions work correctly. Furthermore, the toolbox is highly reviewed within the Matlab community. To use these functions it is still necessary to program the generation of the transition probability tensor and reward matrix ourselves. The following list gives a summarized overview of how the Matlab scripts solve the DTMDP with the given input.

1. First the following variables are used as input: B, K, λ_k, μ_k, θ_{kk+1}, $\delta_{k_{max}}$, p_k, c_{dk}, c_{bk}, γ.

2. A state space matrix is generated, that consists of all possible states the MDP can be in.

3. To keep track of blocked patients and dying patients, it is necessary to add a penalty state space to the original state space, such that it is possible to add a penalty cost for a transition going from the normal state space to the penalty state space.

4. An empty transition probability tensor Q (SXSXA) is generated based on the number of states possible states S and and A the number of possible actions.

5. A transition matrix T (SXS) is generated based on the state space matrix and the given input variables. Each cell 'q(i,j)' with index i being the current state, index j being the next state, is filled with its' corresponding transition rate (event based transition rates).

6. The sum of all outgoing state rates per state is calculated from T. The highest total outgoing rate of a state is then assigned as uniformization rate η or a higher predefined uniformization η rate is used. It is also possible to assign a predefined uniformization rate, as long as it is equal or higher than the highest total outgoing rate. Then Equation (4.2.4.2) is used to change each cell 'q(i,j)' into 'p(i,j)' and thus make the Continous MDP into a Discrete MDP. The cell 'p(i,j)' indicates the probability of going from state i to j.

7. The new transition probability matrix T is added to the transition probability tensor Q. Moreover, the state changes due to assigning a patient in state k are also added to the transition probability tensor Q.

[1] Marie-Josee Cros (2020). Markov Decision Processes (MDP) Toolbox (https://www.mathworks.com/matlabcentral/fileexchange/25786-markov-decision-processes-mdp-toolbox), MATLAB Central File Exchange. Retrieved August 24, 2020.

8. A cost matrix C (SXS) is generated with penalty costs added to all state transitions where a patient dies or is blocked. All infeasible transitions are highly penalized to prevent actions that should not happen (i.e. Treat patient 1, even though there are no patients of class 1.).

9. A MDP check from the MDP toolbox is run to see whether both the transition tensor Q and cost Matrix C are correct in terms of size, all outgoing probability transitions per state adding up to 1 and non-negative probabilities.

10. Then value iteration or policy iteration is run depending on the computation time and the required processing power.

4.3.2 Verification and validation

Verification and validation are necessary to check whether our Matlab model works for its intended purpose. The verification process is the evaluation of whether or not the functions of our Matlab model behave the way we want them to behave. For example, in step 5 of the computational methodology, a transition matrix T is created depending on the input variables and we want to know if those values are correct. Due to the enormous size our state space can be in, depending on the number of beds, it is practically impossible to check for a large number of beds whether the transition probability tensor Q is correct. Therefore, during development, all functions of the Matlab model are tested with a very small number of beds (3) and verified by hand. In addition, the Matlab program also checks for any incorrectness in code and provides us with error feedback. Furthermore, every time an MDP is solved the following check functions are executed for the input tensor Q and cost matrix C:

- Check if the number of rows and column per matrix in tensor Q are equal (i.e. check if the number of in going states is equal to the number of out going states)
- Check if every row sums to 1
- Check if all probabilities are positive
- Check compatibility between tensor Q and cost matrix C, by checking whether the matrices in tensor Q are of equal size as cost matrix C.

The validation process is the assurance that the Matlab model provides us with the answer we are looking for. In other words, is the output policy correct with the input variables. To validate our model, several extreme test cases are made for the 1 server and 2 class model. Four different test cases are depicted in Table 4 with the results in Figure 4.

Table 4: Extreme cases

Case	Description	λ_1	λ_2	μ_1	μ_2	p_1	p_2	θ_{12}	δ_2
0.1	High utilization	20	10	20	10	0.98	0.95	0.5	0.25
0.2	Low utilization	0.2	0.1	20	10	0.98	0.95	0.5	0.25
0.3	High death rate	6	2	20	10	0.98	0.95	0.5	10
0.4	High transfer rate	6	2	20	10	0.98	0.95	10	0.25
B = 10, c_{b1} = -2, c_{b2} = -5, c_d = -5, γ = 0.9999									

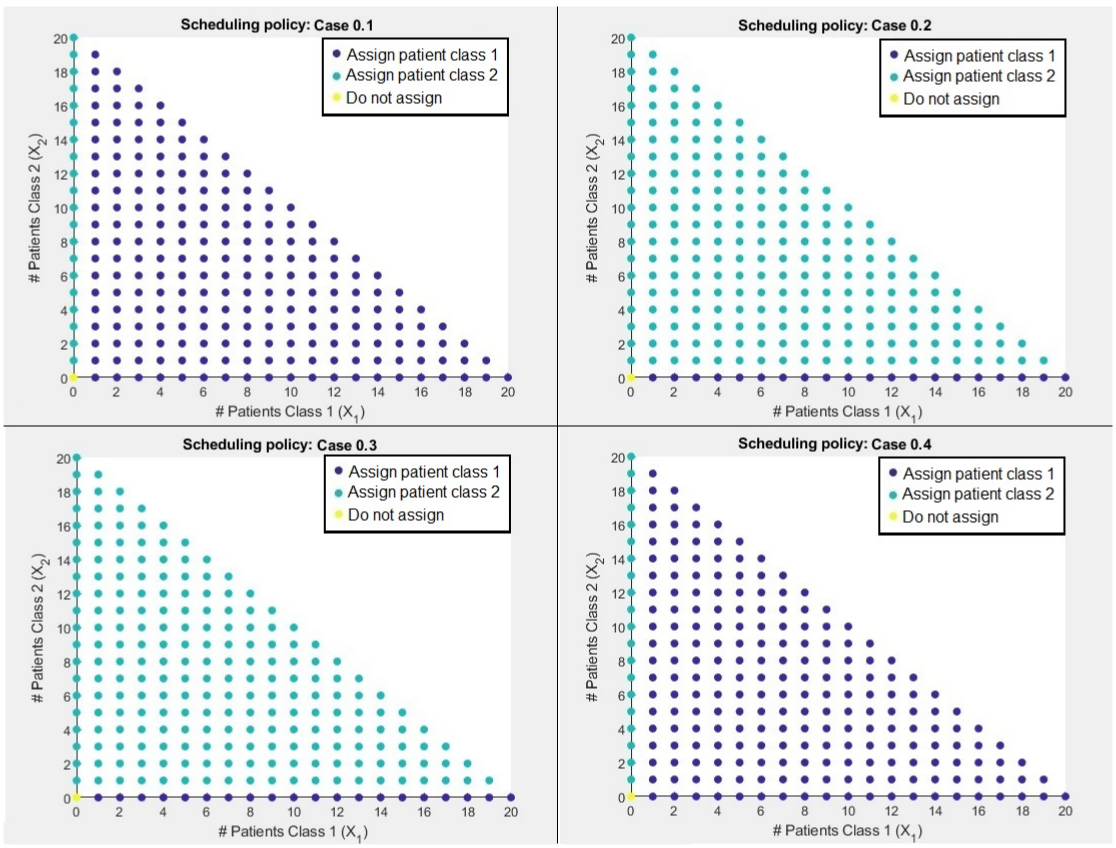


Figure 4: Extreme cases

The results from the extreme cases can be seen in Figure 4. In the scheduling policy, the y-axis depicts the number of priority class 2 patients waiting for treatment and the x-axis depicts the number of patients of priority class 1 patients waiting for treatment. Every circle represents a possible state of the system where the scheduler has to decide who to treat, while the different colors are used to depict the action taken for each state, which can also be seen in the legend. Purple indicates to assign a class 1 patient to an available server for treatment, blue indicates to assign a class 2 patient to an available server for treatment and yellow indicates to not assign. For example in case 0.1, 2 class 1 patients waiting (X_1) and 2 class 2 patients waiting (X_2) indicates to assign a class 1 patient to an available server, while 0 class 1 patients waiting (X_1) and 2 class 2 patients waiting indicates to assign a class 2 patient to an available server.

The policy for the high utilization case 0.1 takes on the form of always treating patient class 1 (the least severe) when both patients of both priority classes are waiting. This is due to the fact that when the arrival rate is very high relative to the treatment rate, then the chance of reaching full capacity and blocking a patient is very likely. Furthermore, a lot of patients in the queue means a lot of patients that can deteriorate. Hence, always treating the least severe if both classes of patients are in the queue is the best option for case 0.1.

In contrast, the policy for the low utilization case 0.2 takes on the form of always treating class 2 patients (the most severe) when both classes of patients are waiting. This is due to the fact that the chance of reaching full capacity and thus blocking a patient is very low. The chances of deteriorating patients are also very low. Hence, in this scheduling policy, the focus is mostly on preventing patients from dying due to deterioration.

For case 0.3, if the mortality rate is relatively very high, which means that a class 2 patients can die very quickly while waiting, then it makes sense to only treat those patients. While in case 0.4, if the deterioration rate is relatively very high, then it makes sense to always treat class 1 patients because otherwise the server mostly needs to deal with class 2 patients only due to the relatively high deterioration rate.

4.3.3 Sensitivity analysis of discount factor

Before we start with the experimental cases it is also important to understand the effect of the discount factor on the policy. The discount factor is related to the net present value calculation. A lower discount factor indicates that earlier decisions are more important then future decisions. The discount factor is applied every time the model takes an action. In our case, due to the uniformization method, an action is taken every fictitious time step with a chance of staying in the same state, dependent on the uniformization relatively to all other outgoing rates. Taking a too low discount factor relative to the uniformization rate might result in an undesirable policy. Hence, to choose a suitable discount factor for all experimental cases of MDP model 1, a sensitivity analysis of the discount factor is executed for the benchmark case 1.1 (see Table 6 of experimental cases 3.2.5). The discount factor is increased until a point is reached where the policy does not change anymore. This happens at the discount factor of 0.9999 and is thus the chosen discount factor. The resulting policies per discount factor are provided in Appendix B.

4.3.4 Computation time

In this section, value iteration and policy iteration are compared against each other on computation time. Furthermore, the number of beds against the computation time is also assessed to determine a suitable number of beds as the number of beds influences the total size of the state space. The experimental cases were implemented in Matlab using a six year old ASUS N751JK Notebook with Intel(R) Core(TM) i7-4710HQ CPU. All computation times are given in Table 5. VI indicates value iteration and PI indicates policy iteration. The chosen number of beds and computation method for each category of experimental cases are underlined. It should be noted that the computation times were based on using Matlab without using any other program. If no computation time was given, then this was either due to the notebook freezing/crashing or computation time taking too long. A cut-off value of 30 secs was chosen as a safe measure due to the fact that a longer running computation time was related to a higher chance of crashing. Hence, the chosen suitable beds are based on putting not too much strain on the notebook, while still being able to show differences in the scheduling policy.

Table 5: Computation time MDP 1

Computation time (s)						
-	M=1	K=2	M=3	K=2	M=1	K=3
Beds	VI	PI	VI	PI	VI	PI
5	1	<1	<1	<1	33	<1
10	2	<1	12	1	283	<u>11</u>
15	16	<1	90	<u>7</u>	-	-
20	60	<u>2</u>	-	32	-	-
25	-	5	-	-	-	-
30	-	12	-	-	-	-
35	-	33	-	-	-	-

4.3.5 Experimental cases

First of all, to make a fair comparison between cases, a predefined uniformization rate is used for all cases. This predefined uniformization rate needs to be equal or higher than the highest outgoing rate of all cases and will be calculated before further analyzing the experimental cases. To answer the first sub-question different experimental cases are set up for the 1 server and 2 patient priority classes in Table 6. 'In.' is short for Increase and 'De.' is short for Decrease. The first case 1.1 is taken as a benchmark to compare against all other cases. Case 1.2, 1.3, and 1.4, 1.5 are based on a difference in arrival rates. The arrival rates in case 1.2 and case 1.3 are respectively halved and doubled. Case 1.4 contains an increase in the arrival rate for class 2 patients and case 1.5 contains an increase in the arrival rate for class 1 patients. As arrival rates can vary heavily, it is important to understand how the scheduling policy reacts to a change in the arrival rates. In case 1.6 the treatment rate is increased. It is expected that an increase in arrival rate relative to the treatment rate leads to a scheduling policy putting more emphasis on treating the least severe patient and vice versa.

In case 1.7 and 1.8, respectively an increase and decrease are made in the transfer rates. In case 1.9 and 1.10, respectively an increase and decrease are made in the mortality rates. As has previously mentioned, these rates are exponentially distributed per patient. This does not indicate at case 1.6 for example that all patients deteriorate from least severe to most severe. For the aforementioned four cases, it is expected that an increasing mortality rate leads to more priority for more severe patients, while an increasing deterioration rate leads to more priority for less severe patients.

Case 1.11 sees a decrease in the probability of successful treatment and the final case 1.12 sees the same probability of successful treatment as case 1.11 with an increase in the mortality rate. The expectation is that if the probability of successful treatment is lower and the difference between the most severe and least severe becomes bigger, then it becomes less attractive to treat a more severe patient due to the probability of successful treatment difference. However, if the mortality rate is increased, it becomes less beneficial to stall a more severe patient from getting treatment.

To answer the second sub-question, four different cases are analyzed for 3 servers and 2 patient priority classes, which can be found in Table 7 with case 2.1 considered as the benchmark case. As has been mentioned previously, a queue can only exist if the scheduler decides to do not assign or all servers are occupied. The first case 2.1 contains a decreased treatment rate compared with the benchmark of case 1.1, as the treatment rate is per server. The relative difference between treatment rates for patient class 1 and patient class 2 is still the same. Case 2.2 contains an increase in the total arrival rate compared to case 2.1. Case 2.3 contains a decrease in transfer rate and case 2.4 contains a decrease in the probability of successful treatment, both compared to case 2.1. Our interest in these four cases is to see how introducing extra servers changes the scheduling policy. The expectation is that knowing which patients are currently being treated affects the scheduling policy, if more than 1 server is introduced in the scheduling problem. This is due to the fact that a decision is always made once a server becomes available. This indicates that in the multiple server cases, if a server becomes available, then other servers might still be treating other patients. In the 1 server cases this is not possible.

Table 6: Experimental cases: 1 server and 2 classes of patients

Case	Description	λ_1	λ_2	μ_1	μ_2	p_1	p_2	θ_{12}	δ_2
1.1	Benchmark	6	2	20	10	0.98	0.95	0.5	0.25
1.2	De. $\{\lambda_1, \lambda_2\}$	3	1	20	10	0.98	0.95	0.5	0.25
1.3	In. $\{\lambda_1, \lambda_2\}$	12	4	20	10	0.98	0.95	0.5	0.25
1.4	In. λ_2	6	6	20	10	0.98	0.95	0.5	0.25
1.5	In. λ_1	10	2	20	10	0.98	0.95	0.5	0.25
1.6	In. $\{\mu_1, \mu_2\}$	6	2	30	15	0.98	0.95	0.5	0.25
1.7	In. θ_{12}	6	2	20	10	0.98	0.95	1	0.25
1.8	De. θ_{12}	6	2	20	10	0.98	0.95	0.1	0.25
1.9	In. δ_2	6	2	20	10	0.98	0.95	0.5	1
1.10	De. δ_2	6	2	20	10	0.98	0.95	0.5	0.1
1.11	De. $\{p_1, p_2\}$	6	2	20	10	0.90	0.70	0.5	0.25
1.12	De. $\{p_1, p_2\}$ & In. δ_2	6	2	20	10	0.90	0.70	0.5	1
B = 20, c_{b1} = -2, c_{b2} = -5, c_d = -5, γ = 0.9999									

Table 7: Experimental cases: 3 servers and 2 classes of patients

Case	Description	λ_1	λ_2	μ_1	μ_2	p_1	p_2	θ_{12}	δ_2
2.1	Benchmark	6	2	8	4	0.98	0.95	0.5	0.25
2.2	In.$\{\lambda_1, \lambda_2\}$	12	4	8	4	0.98	0.95	0.5	0.25
2.3	De. θ_{12}	6	2	8	4	0.98	0.95	0.1	0.25
2.4	De. $\{p_1, p_2\}$	6	2	8	4	0.90	0.70	0.5	0.25
B = 15, c_{b1} = -2, c_{b2} = -5, c_d = -5, γ = 0.9999									

To answer the third sub-question seven cases are analyzed with 1 server and 3 patient priority classes, which can be found in Table 8 with case 3.1 considered as the benchmark case. Case 3.2 and 3.3 vary in different arrival rates, respectively doubled and halved again. Case 3.4 contains an increase in deterioration rate and mortality rate. Case 3.5 contains a decrease in deterioration rate for the first priority class but an increase in mortality rate for the third priority class. Case 3.6 contains a decrease in the probability of successful treatments. Finally in case 3.7, the deterioration rate of class 2 patients is increased and the deterioration rate of class 2 patients is decreased. It is expected that changes in these parameter values result in scheduling policy changes that follow the same changes as for the 1 server and 2 priority cases.

Table 8: Experimental cases: 1 server and 3 classes of patients

Case	Description	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	p_1	p_2	p_3	θ_{12}	θ_{23}	δ_3
3.1	Benchmark	6	3	1	20	15	10	0.99	0.95	0.90	1	0.5	0.25
3.2	In. $\{\lambda_1, \lambda_2, \lambda_3\}$	12	6	2	20	15	10	0.99	0.95	0.90	1	0.5	0.25
3.3	De. $\{\lambda_1, \lambda_2, \lambda_3\}$	3	1.5	0.5	20	15	10	0.99	0.95	0.90	1	0.5	0.25
3.4	In. $\{\theta_{12}, \theta_{23}, \delta_3\}$	6	3	1	20	15	10	0.99	0.95	0.90	2	1	0.5
3.5	De. θ_{12} and In. δ_3	6	3	1	20	15	10	0.99	0.95	0.90	0.5	0.5	0.5
3.6	De. θ_{12} and In. θ_{23}	6	3	1	20	15	10	0.99	0.95	0.90	0.25	1	0.25
3.7	De. $\{p_1, p_2, p_3\}$	6	3	1	20	15	10	0.90	0.80	0.60	1	0.5	0.25
B = 10, c_{b1} = -2, c_{b2} = -5,c_{b3} = -10, c_d = -10, γ = 0.9999													

4.4 MDP model 1: Results

In this section, the findings of the computational cases, that can be found in Chapter 4.3, are reviewed. Chapter 4.4.1 describes the results of the '1 server and 2 priority classes' experimental cases. Chapter 4.4.2 describes the results of the '3 servers and 2 priority classes' experimental cases. Finally, chapter 4.4.3 describes the results of the '1 server and 3 priority classes' experimental cases. Before giving the results of all cases, it should be noted that the predefined uniformization rate that was used for all experimental cases was 50 (i.e. $\eta = 50$). The highest total outgoing rate was found in case 3.2, which was equal to 49.

4.4.1 Results: 1 server, 2 priority classes

The scheduling policies of the first six cases, respectively cases 1.1 until case 1.6, are depicted in Figure 5. The scheduling policies are depicted the same as in chapter 4.3.2 verification and validation. The average cost value V is calculated by taking the average of all value functions from each possible starting state and can be found underneath the legend. The value function contains the discounted sum of the rewards to be earned by following the optimal policy from a starting state. As has been mentioned in chapter 4.2.3 the discounted sum of the rewards in this MDP is the discounted sum of the costs and are therefore negative.

The first thing that can be noticed is that most scheduling policies take on the form of a switching curve, which is shown by a red line. The switching curve splits the optimal policy in two sides. On the left side of the switching curve, it is always optimal to treat patients class 2 (more severe patients) and on the right side of the switching curve it is always optimal to treat patients class 1. This makes sense as the closer we get to maximum capacity, the more preferable it becomes to treat less severe patients with a lower treatment time. It should also be noted that not choosing a patient for treatment (i.e. being idle) is never the case when patients are waiting. It is always optimal to treat a patient if a patient is waiting for treatment. This is true for all the 1 server and 2 priority class cases.

The switching curve of case 1.1 shows a straight diagonal line, which indicates that the only requirement for this policy is the total number of patients in the pre-admission room and is independent of the distribution of waiting patients. In other words if patients of both classes are waiting for treatment and there are more than 12 patients waiting then it is optimal to treat a class 1 patient. The scheduling policy of case 1.2 (decrease in both arrival

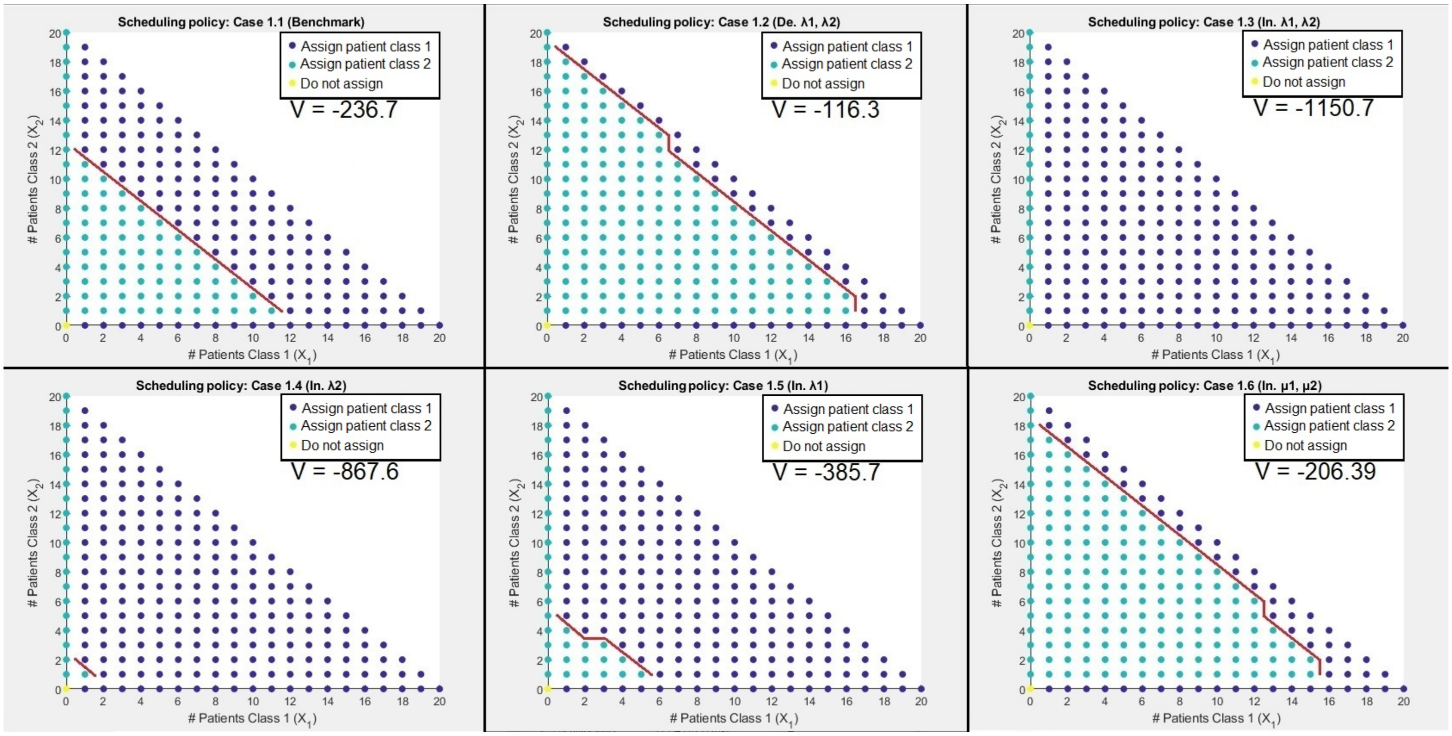


Figure 5: Scheduling policies: case 1.1-case 1.6

rates) shows a switching curve more to the right in comparison with case 1.1. A lower arrival rate ensures that there is more room to treat more severe patients as the odds of capacity reaching its limit and thus block other patients is much lower. Besides the switching curve moving, the switching curve does not take on the form of a diagonal straight line. For example, if 1 class 1 patient and 18 class 2 patients are waiting for treatment, then it is optimal to assign a class 1 patient for treatment. Furthermore, if 1 class 2 patient and 18 class 1 patients are waiting for treatment, then it is optimal to assign a class 2 patient for treatment. Even though in both situations the total number of patients are waiting for treatment is the same, a difference exists in who should be treated first. Since a situation with more class 2 patients waiting than class 1 patients translates in higher odds of someone dying and thus higher odds that a bed becomes available. For case 1.3, the switching curve is nonexistent and comes down to always prioritize treating the least severe patients, if both classes of patients are waiting for treatment. In this case, it is much more likely that all beds are occupied and patients have to be blocked from the healthcare department. Therefore, priority lies on choosing the patient that requires the least treatment time. Moreover, it can be seen that a lower arrival rate results in less average costs and a higher arrival rate results in more average costs. In case 1.4 with equal arrival rates, the switching curve moved to the left in comparison with case 1.1. The same reasoning as for case 1.3 can be applied here. In case 1.5, the switching curve moved to the left due to the increase in arrival rate. The total arrival rate for case 1.5 is the same as for case 1.4. However, the switching curve is different, due to the fact that class 1 patients require less treatment time. Moreover, something that can also be noticed here is that a horizontal line can be found in the switching curve. The switching curve of case 1.6 moved to the right in comparison with case 1.1. The same reasoning as in case 1.2 can be applied here. As increasing the treatment rate leads to a same effect as decreasing the arrival rate in terms of utilization.

The scheduling policies of the other six cases, respectively cases 1.7 until case 1.12, are depicted in Figure 6. Let us first take a look at case 1.7 and case 1.8, respectively increasing and decreasing the deterioration rate from priority class 1 to class 2. Not surprisingly, increasing the deterioration rate in case 1.7 moved the switching curve more to the left side. This is because to prevent having more patients from class 2 and thus have more work, it becomes more optimal to treat patients from class 1 when fewer beds are occupied. The opposite is true for case 1.8, where the deterioration rate is decreased. The switching curves of case 1.9 and case 1.10 are also as expected. Respectively increasing the mortality rate leads to the switching curve moving more to the right and decreasing the mortality rate moves the switching curve to the left, as a higher mortality rate value leads to higher odds of a dead patient and thus a penalty cost. The reason of the vertical lines in case 1.8 and 1.9 are similarly explainable as in case 1.2. The more class 2 patients that are waiting for treatment, the higher probability that someone might die. If a patients dies, then this automatically results in a bed becoming available again. Hence, when there are a lot of class 2 patients waiting for treatment, it is very likely that a bed will become available due to the death. Therefore, the vertical lines can be explained by the relative difference of the mortality rate and the deterioration rate. Case 1.11 does not contain a switching curve and is similar to case 1.3. It basically comes down to always treating the least severe patient when both classes of patients are waiting. In case 1.12, the mortality rate is increased, and makes it less beneficial to stall class 2 patients, therefore a switching curve can be seen in comparison with case 1.11.

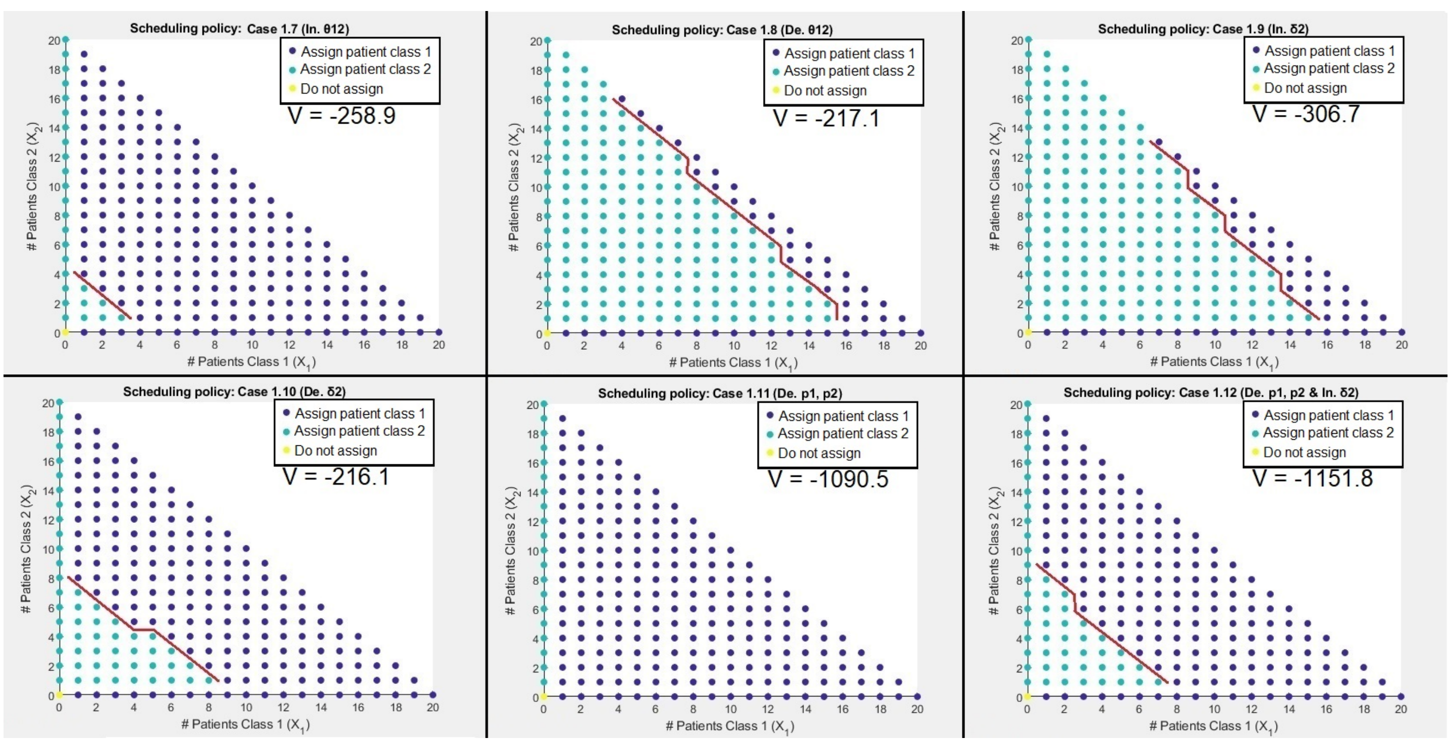


Figure 6: Scheduling policies: Cases 1.7-1.12

4.4.2 Results: 3 servers, 2 priority classes

In the previous section, ten different cases were depicted to show how the scheduling policy moves depending on changing parameter values. In this section, we will take a look at the scheduling policy, while using multiple servers. Figure 7 shows a scheduling policy for multiple servers. It takes on the form of multiple scheduling policies in one figure. The y-axis denotes the number of class 2 patients in treatment and the x-axis denotes the number of class 1 patients in treatment. As 1 server can always treat 1 patient at a time, then this also means that the y-axis denotes the number of servers currently treating class 1 patients and the x-axis denotes the number of servers currently treating class 2 patients. The patients currently in treatment can also be read from the title of each scheduling policy. Server distribution [class 1 patients currently in treatment, class 2 patients currently in treatment] indicates what patients are currently being treated by the servers. For example, if 1 class 1 patient and 1 class 2 patient are currently in treatment, then the reader should look at the scheduling policy with title server distribution [1,1]. The scheduling policy for each server distribution can be read in the same manner as in the previous single server cases. The scheduling policy for case 2.1 and case 2.3, can be seen in this chapter and the less interesting scheduling policies for case 2.2 and 2.4 can be seen in Appendix C.

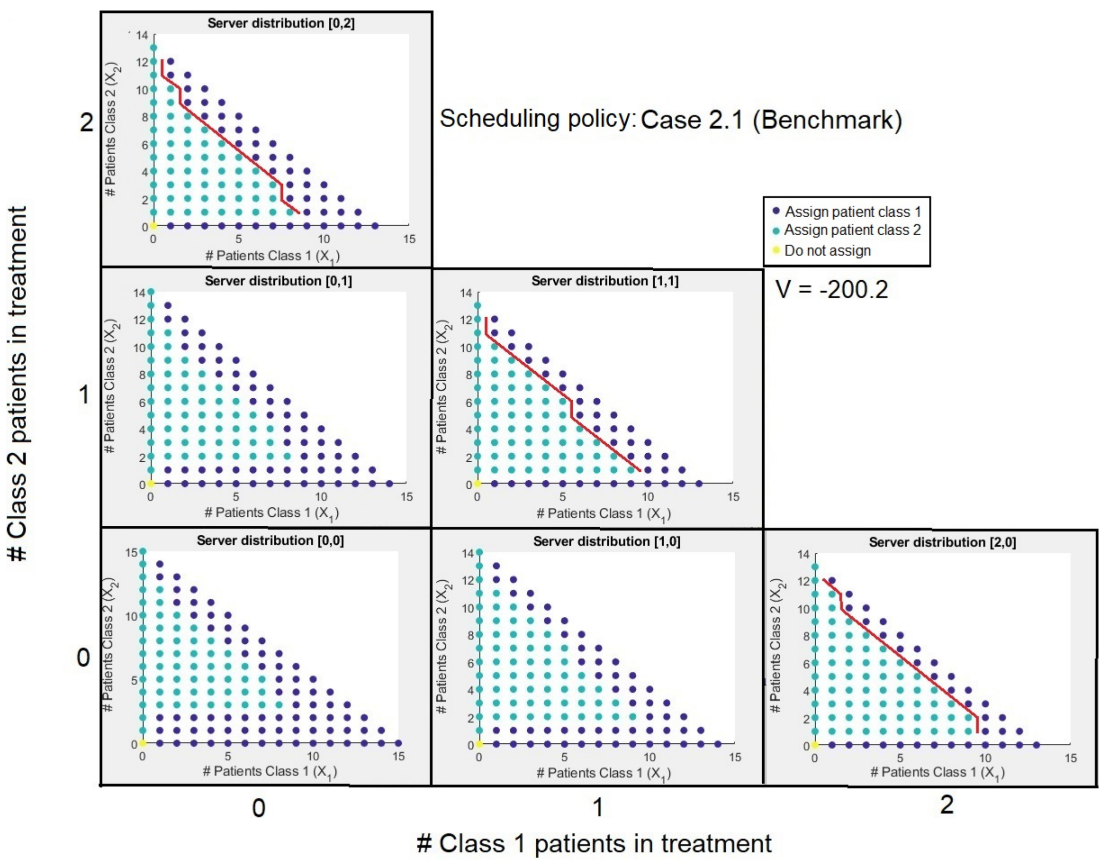


Figure 7: Scheduling policy: case 2.1

There are two interesting things to note here. First, it was assumed that no time elapses with choosing a a patient for treatment and moving a patient from the pre-admission room to the treatment room. Therefore, it is impossible that a different event (e.g. the arrival of another patient) happens in between choosing two patients for treatment. It is also impossible that two servers at the same time are available, except if idling with more than 1 server would be an optimal action. In the case of Figure 7, this means that the scheduling policies where more than 1 server is available are unnecessary, because it can never be the case that a queue exists if more than 1 server is available. The second interesting thing to note here is that each scheduling policy per server distribution takes on the form of a switching curve and that a difference exists between switching curves. In Figure 7, there are 3 different server distributions for 2 occupied servers: [0,2] [1,1] and [2,0]. Even though the same number of servers are occupied, the switching curve differs. This shows that the scheduling policy changes depending on what patients are currently in treatment. This makes sense, because if there are a lot of very severe patients being treated, then the odds of a bed/server becoming free are less likely than when a lot of less severe patients are being treated.

For case 2.2 the scheduling policy's switching curves moves more to the left due to an increase in arrival rate. This is in line with the resulting difference between case 1.2 and case 1.1. Furthermore, the two interesting things that were mentioned for case 2.1 also apply to this case. This scheduling policy can be found in Appendix C.

The scheduling policy for case 2.3 is depicted in Figure 8. It can be noticed that in this scheduling policy, deciding to not treat a patient is the most optimal action for the final server under certain circumstances. For example, if 2 servers are currently treating 2 class 1 patients, then it is optimal to not assign the final server to a class 1 patient if less than 5 class 1 patients and exactly 0 class 2 patients are waiting for treatment. This is due to the fact that the lower deterioration rate and the use of multiple servers makes it more attractive to stall less severe patients in case a more severe patient comes in. It should also be noted that when 2 servers are currently treating 2 class 2 patients, then not assigning the final server is only optimal if less than 4 class 1 patients and exactly 0 class 2 patients are waiting for treatment. This also shows that the idling (not assigning) strategy differs per server distribution.

For case 2.4 the scheduling policy's switching curves moves more to the left due to the lower probability in successful treatment. However, this switching curve is only apparent in the [1,1] and [0,2] server distributions.

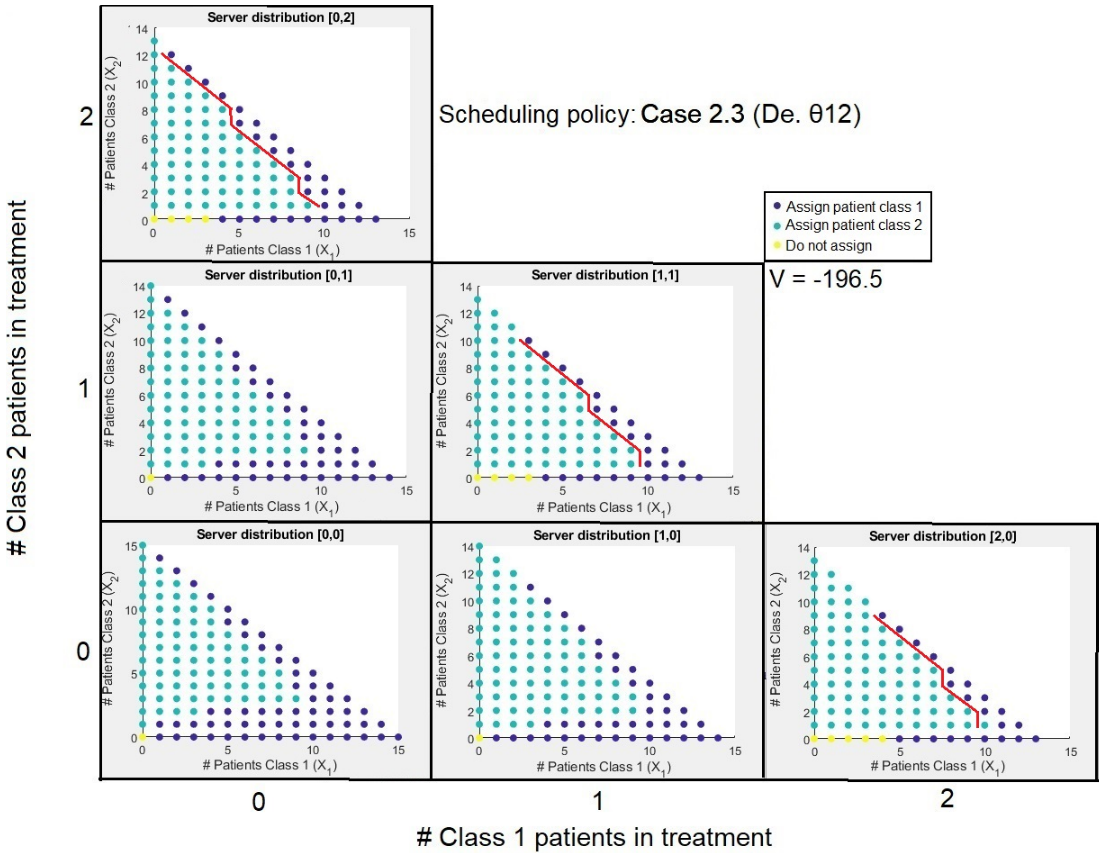


Figure 8: Scheduling policy: case 2.3

4.4.3 Results: 1 server, 3 priority classes

In this section, the 1 server and 3 priority class scheduling policies will be discussed. Figure 9 shows the scheduling policies for case 3.1, 3.2, 3.3 and 3.4. Figure 10 shows the scheduling policies for case 3.5, 3.6 and 3.7. The scheduling policy of 3 priority classes takes on the form of a 3 dimensional scatter plot. The x-axis depicts the number of class 1 patients waiting for treatment. The y-axis depicts the number of class 2 patients waiting for treatment. The z-axis depicts the number of class 3 patients waiting for treatment. There are 4 action dots: purple, blue, green, and yellow. Respectively they indicate assign patient class 1, assign patient class 2, assign patient class 3 and do not assign.

Let us first take an overall look at all scheduling policies. The changes in the scheduling policies due to the changing parameters follow the same reasoning as for the 1 server and 2 priority class experimental cases. For example, increasing the arrival rate, makes less severe patients be prioritized more often. The scheduling policy can basically be split up into three different scenarios. The first scenario is when there are only patients waiting from one particular class. All scheduling policies show that when there is a single class of patients waiting for treatment, then it is always the best action to treat that particular class of patients and to never keep a server idle.

The second scenario is when two classes of patients are waiting. This scenario can show a switching curve between those two classes of patients. For example in case 3.1, it can be seen that between class 3 patients and class 2 patients a switching curve exists, similarly to the switching curve from the 2 class, 1 server model.

The third scenario is when at least one patient is waiting from each patient class. In this scenario, the action is mainly to either treat a class 1 patient or a class 3 patient. This can be explained by the fact that there are penalty costs involved for letting someone die and for blocking a patient. Class 3 patients are the closest to death, while class 1 patients have the highest treatment rate. Therefore if capacity is becoming a problem, then it is always a better option to immediately switch from class 3 patients to class 1 patients. However, the scheduling policy of case 3.6 shows that in certain states it is optimal to treat class 2 patients, due to the relatively high deterioration rate of class 2 patients.

Now lets explain one scheduling policy in more detail. In case 3.1 it can be seen that a switching curve exists between the blue and green dots when $X_3 > 0, X_2 > 0, X_1 = 0$, but no switching curve exists between the purple and blue dots when $X_3 = 0, X_2 > 0, X_1 > 0$. When $X_3 > 0, X_2 > 0, X_1 > 0$, the only dots that can be seen are green dots (assign the most severe patients) and purple dots (assign the least severe patient). If we compare it with case 3.2 with increased arrival rates it is expected that assigning less severe patients becomes more important, while decreasing the arrival rates as in case 3.3 makes the most severe patients more important. In case 3.4, the scheduling policy barely changes in comparison with case 3.1, which can be expected as the relative difference between both deterioration rates and mortality rate stays the same. By decreasing the deterioration rate of the least severe patients as in case 3.5 and 3.6, the class 2 patients become more important in the scheduling policy. In case 3.7, it can be seen that decreasing the probability of successful treatment leads to always prioritizing the least severe.

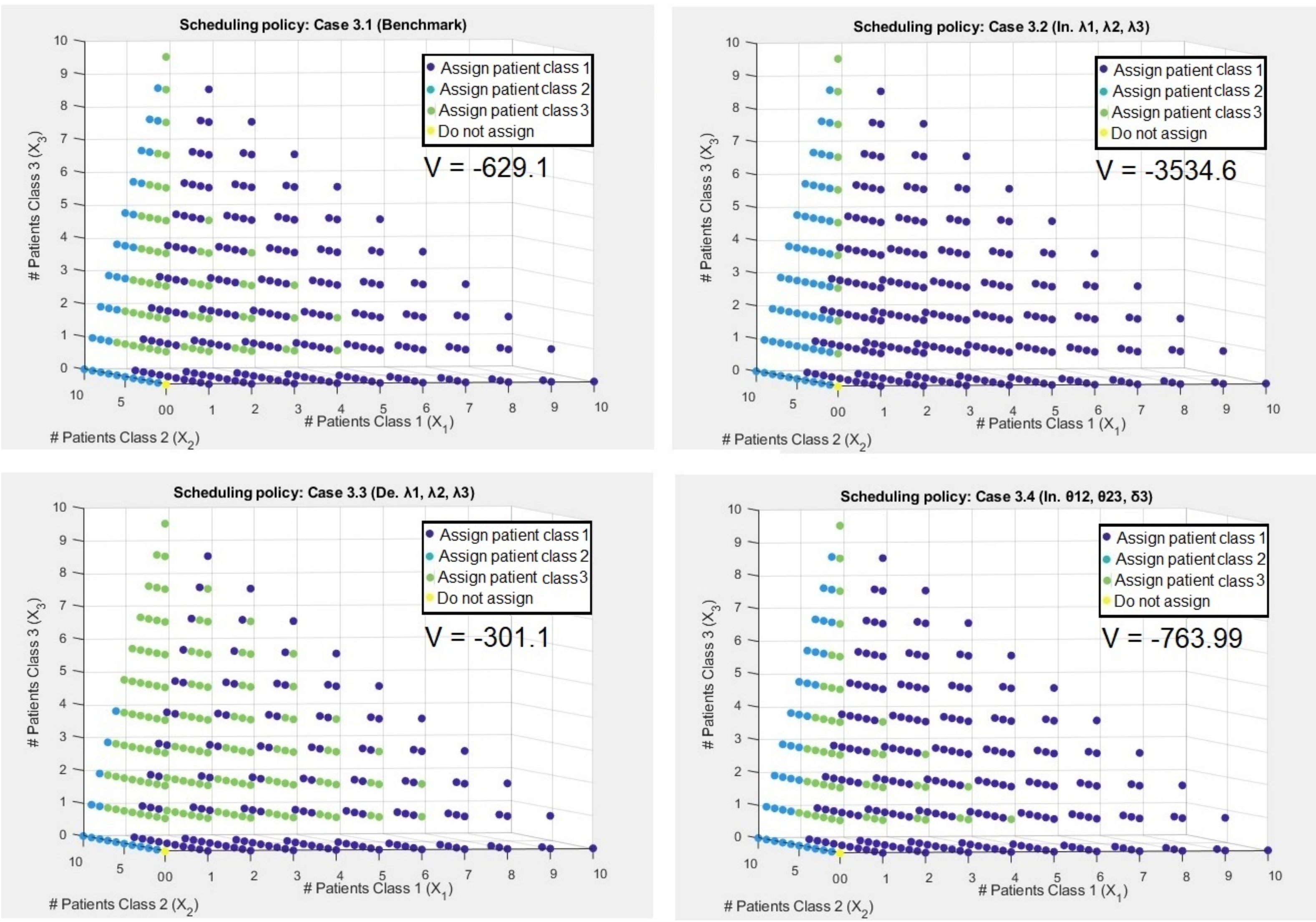
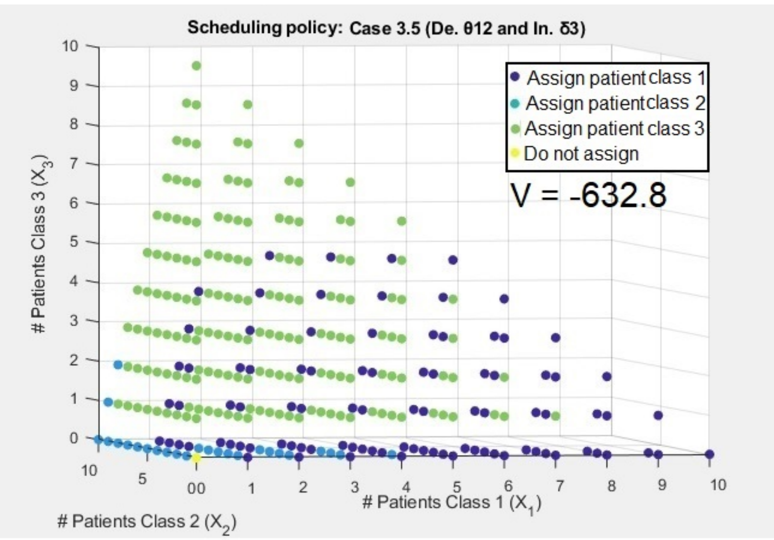
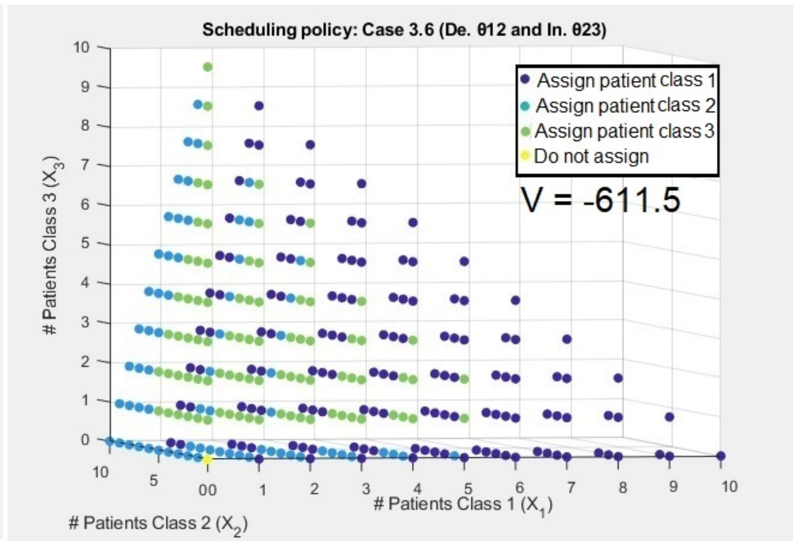


Figure 9: Scheduling Policies: Cases 3.1-3.4





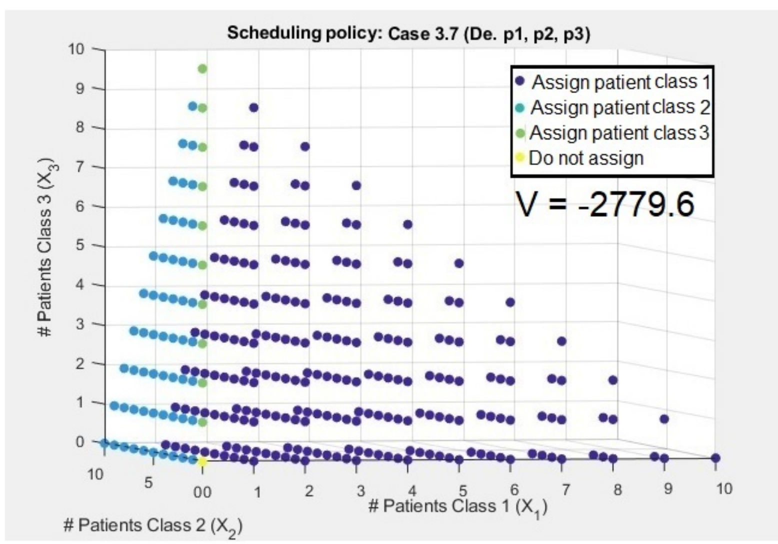


Figure 10: Scheduling Policies: Cases 3.5-3.7

5 Model 2: Treating patients with internal recovery

The second model "Model 2" for treating patients without internal recovery will be discussed in this chapter. This model extends MDP model 1 by taking into account recovery at the healthcare department. The first subsection gives a description of the situation of model 2. The second subsection provides the mathematical formulation of a Markov Decision Process(MDP) model. It should be noted that the cost function, action space and solving methods are exactly the same as in MDP model 1 and are therefore not provided in this subsection. The third subsection describes the computational methodology and the final subsection describes the results of the second MDP model.

5.1 Description

5.1.1 Process of treating a patient with internal recovery

Figure 11 shows a schematic representation of the new situations, which has a small difference in comparison with the schematic representation in chapter 4.1.1 of MDP model 1. The total recovery rate, which is dependent on the number of class k patients in recovery, is added to the model.

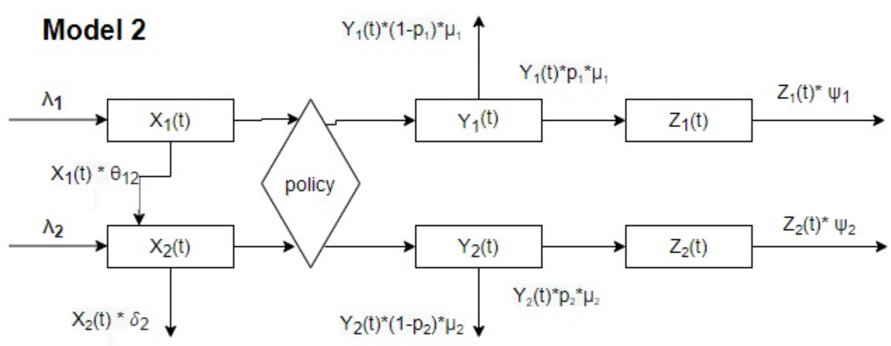


Figure 11: Process of treating a patient for 1 server and 2 patient classes with internal recovery

Patient recovery After treatment a patient will have to recover in the healthcare department, occupying the same bed the patient had since the start. A patient's recovery time ψ_k is modeled with an exponential distribution based on the condition of the patient at the start of the treatment. If the recovery time has elapsed, then the patient will be discharged from the healthcare department and his bed will become available for a new patient. It is assumed that a patient can not die while recovering. $Z_k(t)$ depicts the number of patients in condition k in treatment at time t. The total recovery rate can be calculated as follows: $Z_k(t) * \psi_k$. Table 9 depicts the added variables.

Table 9: Added variables list

Variable	Description
$Z_k(t)$	Number of patients in state k recovering at time t
ψ_k	Recovery rate of each patient in state k after treatment

5.2 MDP Formulation

5.2.1 State space

The state space is finite and dependent on the total number of beds B, the total number of servers M and the number of priority classes K. The state of the system at time t can be described by the number of patients per class waiting for treatment, in treatment and recovering from treatment. Let $X_k(t)$ denote the number of patients waiting for treatment, $Y_k(t)$ denote the number of patients in treatment and $Z_k(t)$ denote the number of patients recovering per class k at time t, then the state at time t can be described as in Equation (5.2.1.1). Furthermore, all possible states are constrained by the number of beds B as in Equation (5.2.1.2).

$$s(t) = (X_1(t), X_2(t), X_k(t), ..., X_{k_{max}}(t); Y_1(t), Y_2(t), Y_k(t), ..., Y_{k_{max}}(t); Z_1(t), Z_2(t), Z_k(t), ..., Z_{k_{max}}(t)) \tag{5.2.1.1}$$

$$\text{s.t.} \quad \sum_{k=1}^{k_{max}} X_k(t) + Y_k(t) + Z_k(t) \leq B \tag{5.2.1.2}$$

The state space S consists of all possible distributions of beds over the priority classes and the number of working servers. The number of possible distributions of beds is quite similar to Equation (4.2.1.3), except that beds can also be assigned to a patient in recovery. This effectively increases the number of the queue's the beds can be divided over. The following formula shows how an increase in the number of priority classes and the number of beds influences the size of the state space S. Let B denote the total number of beds and K the number of priorities, then the total number of states can be calculated as in Equation (5.2.1.3).

$$\sum_{b=1}^{M} \frac{(b + 3k_{max} - 1)!}{b!(3k_{max} - 1)} + \sum_{b=M+1}^{B} (\frac{(b + 2k_{max} - 1)!}{b!(2k_{max} - 1)} * \sum_{b=1}^{M} \frac{(b + k_{max} - 1)!}{b!(k_{max} - 1)}) \tag{5.2.1.3}$$

5.2.2 Uniformization from CTMDP to DTMDP

This section is exactly the same as Chapter 4.2.4 in MDP model 1, except for the calculation of the uniformization rate. The recovery of patients is now also included in the uniformization rate, which can be seen in Equation(5.2.2.1). This is constrained by Equation(5.2.1.2)

$$\eta = \max_i q_i = \max_{Xk,Yk\,\forall k \in K} [\sum_{k=1}^{kmax} (\lambda_k + \psi_k * Z_k + \mu_k * Y_k) + \sum_{k=1}^{kmax-1} (X_k * \theta_{kk+1}) + X_{kmax} * \delta_{kmax}] \tag{5.2.2.1}$$

5.2.3 Transition probabilities

The transition probabilities of model 2 are based on the following events: Arrival of a patient, successful treatment of a patient, failed treatment (death) of a patient, deterioration of a patient, death of a waiting patient and recovery of a patient. In comparison with model 1, only recovery of a patient is added. However, some of the state changes are different in comparison with model 1, due to the added state space variable $Z_k(t)$. Table 10 will follow the same depiction as Table 3 from Chapter 4.2.5. Again, assign patient state k is only possible when patients are waiting for treatment and a server is free.

Table 10: Transition probabilities

Events	$Pr(s_{t+1}\|s_t)$	$s(t) \rightarrow s(t+1)$
Arrival of patient k	$\frac{\lambda_k}{\eta}$	$X_k + 1$
Successful treatment patient k	$\frac{\mu_k * p_k}{\eta}$	$Y_k - 1, Z_k + 1$
Failed treatment patient k	$\frac{\mu_k * (1-p_k)}{\eta}$	$Y_k - 1$
Deteriorate patient k	$\frac{\theta_{kk+1} * X_k}{\eta}$	$X_k - 1, X_{k+1} + 1$
Death patient k_{max}	$\frac{\delta_{kmax} * X_{kmax}}{\eta}$	$X_{k_{max}} - 1$
Recovered patient k	$\frac{\psi_k * Z_k}{\eta}$	$Z_k - 1$
Actions		
Assign patient state k	-	$X_k - 1, Y_k + 1$

5.3 MDP model 2 computational experiments

First of all the computational methodology follows the same process as mentioned in MDP Model 1. The verification process also follows the same procedure as mentioned for MDP Model 1. The first subsection contains a cross validation between MDP model 1 and 2. The second subsections gives the computation time and the third subsection gives an overview of the experimental cases.

5.3.1 Cross Validation

To validate the extension model, we can compare model 1 with model 2. The cases for both models are depicted in Table 11. All variables of case 4.2 are the same as case 4.1, except that a very high recovery rate is included. A very high recovery rate relative to the other values means that the recovery time for a patient is very short, almost non existent. It is expected that the scheduling policies in both cases are exactly the same and the average costs V for both models should be very close to each other. To have an equal comparison the uniformization rate for both cases should be the same. Moreover, the discount rate should be increased as well due to the heavily increased uniformization rate. The highest outgoing total transition rate in case 4.2 is equal to 4008. Therefore the chosen uniformization rate for both cases is 4008. Moreover, as this uniformization rate is approximately a hundredfold of the used uniformization rate in MDP model 1 analysis, the discount rate will be 0.999999. The scheduling policies as seen in Figure 12 are as expected.

Table 11: Cross validation cases

Case	Description	λ_1	λ_2	μ_1	μ_2	ψ_1	ψ_2	p_1	p_2	θ_{12}	δ_2
4.1	Model 1	6	2	20	10	-	-	0.98	0.95	0.5	0.25
4.2	Model 2	6	2	20	10	400	200	0.98	0.95	0.5	0.25
beds = 10, c_{b1} = -2, c_{b2} = -5, c_d = -5, γ = 0.999999											

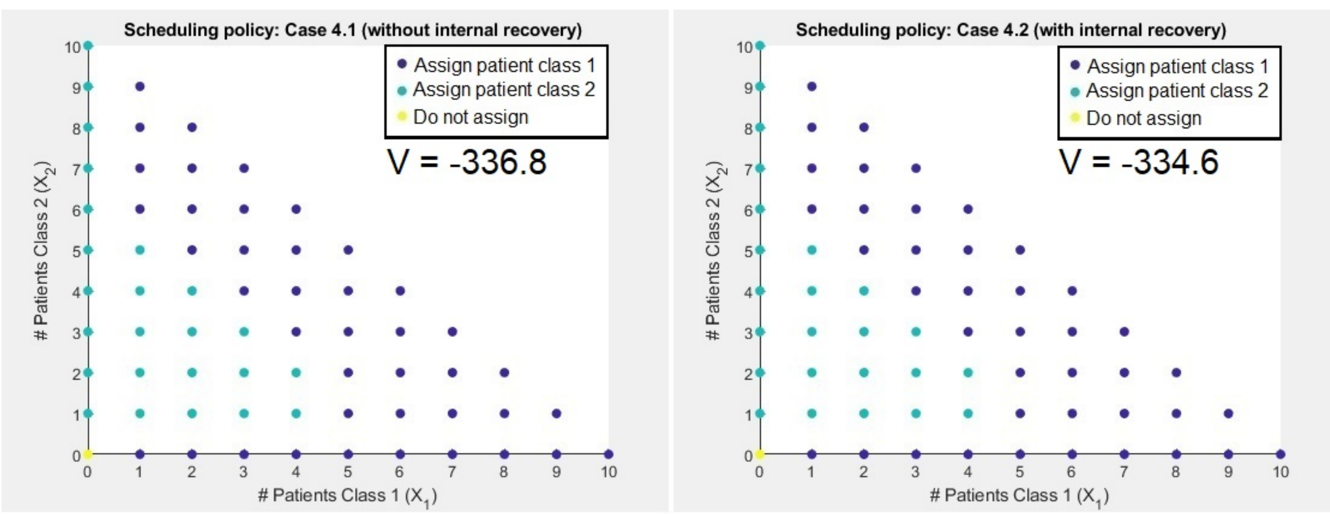


Figure 12: Cross validation

5.3.2 Computation time

Including an extra variable to the state space can have an effect on the computation time and the needed processing power. Therefore, in this section, value iteration (VI) and policy iteration (PI) are both measured again in terms of computation time. Even though it is expected that policy iteration will have the upper hand, it it still useful to know a suitable number of beds that can be used for analysis. All computation times are given in Table 12.

Table 12: Computation time MDP 2

Computation time (s)						
-	M=1	K=2	M=3	K=2	M=1	K=3
Beds	VI	PI	VI	PI	VI	PI
2	<1	<1	<1	<1	<1	<1
4	<1	<1	<1	<1	6	<1
6	2	<1	27	1	270	41
8	20	1	-	-	-	-
10	95	6	-	-	-	-
12	-	57	-	-	-	-

5.3.3 Experimental cases

For the experimental cases including recovery, we are mostly interested in how including recovery changes our priority policy. As this model is basically an extension of MDP model 1, it feels unnecessary to do a very exhaustive research for each parameter. Furthermore, including multiple servers and more than 2 priorities severely decreases the number of beds that can be researched and it is expected that the same conclusions can be drawn for this model as has been suggested in MDP model 1. Therefore they are left out of the experimental cases for this part. All three cases can be found in Table 17. The first case, case 5.1 is similar to case 1.1, but this time with the recovery rate included. In case 5.2, the treatment rate is increased and in case 5.3 the recovery rate is decreased in comparison to case 5.2.

Table 13: Experimental cases

Case	Description	λ_1	λ_2	μ_1	μ_2	ψ_1	ψ_2	p_1	p_2	θ_{12}	δ_2
5.1	Benchmark	6	2	20	10	20	10	0.98	0.95	0.5	0.25
5.2	In. μ_1 and μ_2	6	2	40	20	20	10	0.98	0.95	0.5	0.25
5.3	De. ψ_1 and ψ_2	6	2	40	20	10	5	0.98	0.95	0.5	0.25
beds = 10, c_{b1} = -2, c_{b2} = -5, c_d = -5, γ = 0.99999											

5.4 MDP model 2 results

First of all the highest uniformization rate of all three recovery cases was calculated from case 5.2, which was equal to 228. Therefore, the chosen uniformization rate for these three recovery cases was equal to 500, which is tenfold of the uniformization rate for MDP model 1. Therefore, the discount rate will be 0.99999. The scheduling policy with recovery included follows the same depiction as the scheduling policies provided in MDP model 1. Figure 13 shows the scheduling policy of case 5.1. A new green dot is added to the scheduling policy, which depicts that the optimal action is not only dependent on the patient distribution waiting for treatment, but also on the patient distribution in recovery. If it is a yellow, purple or blue dot, then this indicates that the chosen priority is only dependent on the patients waiting for treatment and not those in recovery. A recovery distribution table that belongs to the scheduling policy of case 5.1 is depicted in Table 14 and shows the conditions under which it is optimal to treat patients of priority class 1. A full table can also be found in Appendix D Table 16. If the condition displayed in Table 14 is not true, then it is optimal to treat a patient of priority class 2. For example, if 2 class 1 patients and 3 class 2 patients are waiting for treatment, then it is only optimal to treat a class 1 patient if the number of patients in recovery for class 2 patients is equal to 5. It should be noted that the second row in Table 14 has the same number of patients waiting in the pre-admission room, but the recovery dependent condition is different. There is one more class 1 patient and one less class 2 patient in the second row of Table 14, but now it is optimal to assign a class 1 patient if there are more than 4 class 2 patients recovering.

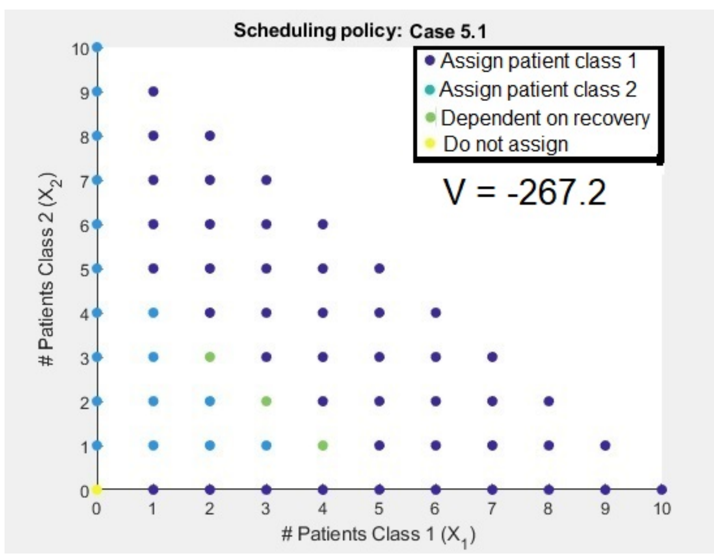


Figure 13: Scheduling policy: case 5.1

For case 5.2, with an increased treatment rate, the number of green dots is increased, which can be seen in Figure 14 on the left side. The increase in green dots can be explained as follows. As recovery is now part of the healthcare department, it takes longer for a bed to become available. The total time it takes for a bed to become free is the sum of the treatment time and recovery time. If the treatment rate increases relative to the recovery rate, then recovery becomes a more important point of congestion than the treatment. Therefore,

Table 14: Recovery distribution table: case 5.1

X_1	X_2	Assign class 1 patient if
2	3	$Z_2 = 5$
3	2	$Z_2 \geq 4$
4	1	$Z_2 \geq 3 \lor (Z_1 \geq 2 \land Z_2 = 2)$

whether the scheduling policy is more or less dependent on the recovery distribution of beds is mostly based on the relative difference between the treatment rate and recovery rate. The recovery distribution table for case 5.2 is a lot bigger and more complex and therefore put in Appendix D Table 17. Due to the high number of possible distributions, a full table is given. For case 5.3 in Figure 14 on the right side, it can be seen that making the relative difference between the treatment and recovery rate even bigger, results in the recovery distribution becoming even more important. As the point of the importance of the recovery distribution has already been proven and the recovery distribution table has to be extracted manually, the exact recovery distribution table is left out.

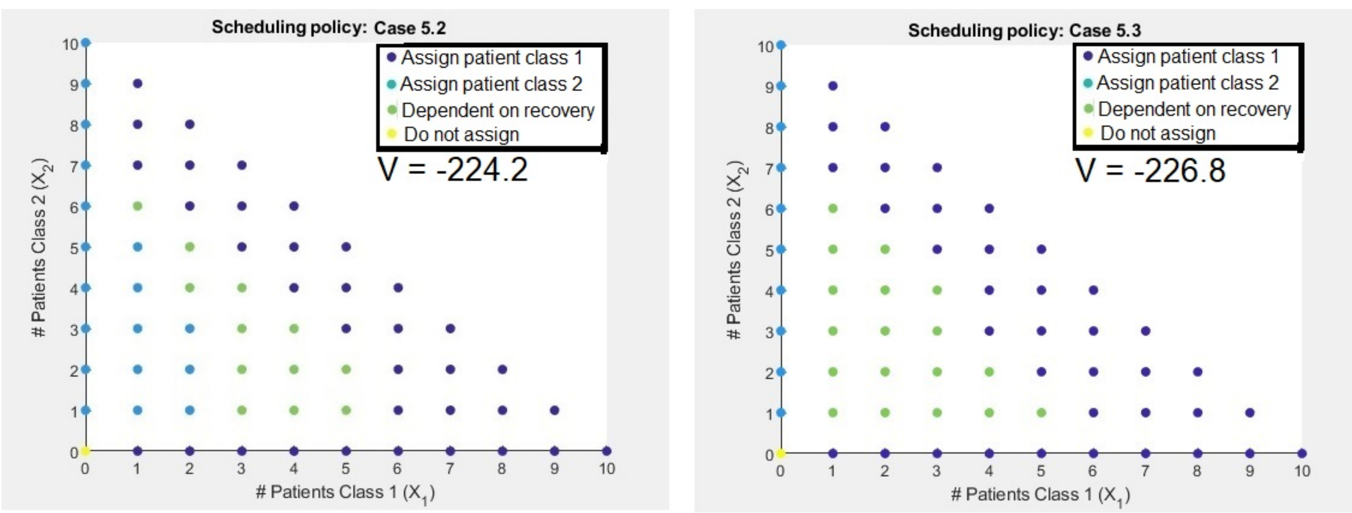


Figure 14: Scheduling Policies: Cases 5.2 and 5.3

6 Comparison against other policies

In the previous results of MDP models 1 and 2, it was shown how the optimal scheduling policy behaves under different circumstances. In this section, a comparison between simple heuristics and the optimal scheduling policy will be provided in terms of the relative difference of the average costs V. The relative percentage difference is equal to (Heuristic average costs V - Optimal average costs V)/Optimal average costs V. The following policies are compared against the optimal scheduling policy:

- LSF: This policy always prioritizes the least severe patient class with the highest treatment rate.
- MSF: This policy always prioritizes the most severe patient class with the lowest treatment rate.
- CAP25: This policy uses the MSF policy if the number of used beds is lower than 25% of the total beds. If the number of used beds is higher than or equal to 25%, then the policy uses the LSF policy. This is basically a simpler version of a state-based policy. The CAP policy is based on the fact that the more patients are occupying a bed in the healthcare department, the bigger chance there is for a patient's arrival to be blocked.
- CAP50: Same policy as CAP25, except the cutoff value is 50% of the bed capacity.
- CAP75: Same policy as CAP75, except the cutoff value is 75% of the bed capacity.

6.1 Experimental Cases

To compare the optimal policy against the simple heuristics, the following six experimental cases were devised. Table 15 shows all experimental cases. Case 6.1 is based on the original benchmark case 1.1 with 1 server and 2 priority classes. Case 6.2 is with an increased mortality rate and decreased deterioration rate. Case 6.3 is based on the original benchmark case 2.1 with 3 servers and 2 priority classes. Case 6.4 is based on the benchmark case 3.1 with 1 server and 3 priority classes. For the final case 6.5 with recovery included, the recovery rate is made relatively low compared to the treatment rate. All experimental cases will be ran with a varying arrival rate, starting with $\lambda_1 = 1.5$ and $\lambda_2 = 0.5$ and incremented with the same value until $\lambda_1 = 15$ and $\lambda_2 = 5$. For the three priority class case, the arrival rate for patient class 3 starts at $\lambda_3 = 0.25$ and increments with the same value until $\lambda_3 = 2.5$.

Table 15: Experimental cases for policy comparison

Case	B	M	k_{max}	μ_1	μ_2	μ_3	p_1	p_2	p_3	θ_{12}	θ_{23}	δ_3	ψ_1	ψ_2
6.1	20	1	2	20	10	-	0.98	0.95	-	0.5	-	0.25	-	-
6.2	20	1	2	20	10	-	0.98	0.95	-	0.25	-	0.5	-	-
6.3	15	3	2	8	4	-	0.98	0.95	-	0.5	-	0.25	-	-
6.4	10	1	3	20	15	10	0.99	0.95	0.90	1	0.5	0.25	-	-
6.5	10	1	2	40	20	-	0.98	0.95	-	0.5	-	0.25	10	5
c_{b1} = -2, c_{b2} = -5, *c_{b3} = -10, c_d = -5 (-10)*, γ = 0.99999, * = only for case 6.4														

6.2 Results

Let us first take a look at both results for the 1 server and 2 patient classes case. Figure 15 shows all resulting cases. The y-axis depicts the percentage difference in performance in terms of average costs. Please do take note, that the scale of the y-axis differs between comparisons and therefore should be read carefully, before interpreting the results. The x-axis depicts the patient arrival rate. For example [1.5; 0.5] means arrival rate of 1.5 for class 1 patients and 0.5 for class 2 patients.

At a very low arrival rate for case 6.1, all policies perform equally well, but this is mostly since the arrival rate is this low that the queue is barely congested. In case 6.2 at a very low arrival rate, LSF performs worse than the other two policies, since class 2 patients have a higher mortality rate than in case 6.1. From both cases, it can be noted that that the MSF and CAP perform near-optimal at low arrival rates, as expected. However, as the arrival rate increases to the higher numbers, MSF and CAP start to worsen in performance. The point where the performance of MSF and CAP starts to worsen is also the point where the performance of LSF starts to improve. The performance of MSF and CAP worsen until a certain value of arrival rate is reached (for example $\lambda_1 = 12$ and $\lambda_2 = 4$ in case 6.1). After this point, the performance of CAP and MSF starts to improve in comparison with the optimal scheduling policy. This can be explained by the fact that if you keep increasing the arrival rate, then the circumstances are so bad that a scheduling policy starts to have less effect on the overall costs. Most interesting is the area underneath the MSF line and LSF line, where both policies do not perform near-optimal, indicating the importance of a state-based policy. CAP, which is a simpler state-based policy, always performs better than MSF.

Increasing the number of servers or priorities as in case 6.3 and 6.4 follows approximately the same plot as in case 6.1. However, between cases 6.1, 6.3, and 6.4 a difference exists between the relative percentage performances. This might be since the number of beds varies between cases. The most interesting to note about case 6.5 is the performance of CAP always being near-optimal. However, this might be due to the limited arrival rate and the increased treatment rate, the experimental case has been tested in. Further, increasing the arrival rate might have shown the same plot as the other four cases. All in all, it can be mentioned that knowing when to shift from a LSF strategy to MSF strategy might also suffice as a reasonable heuristic.

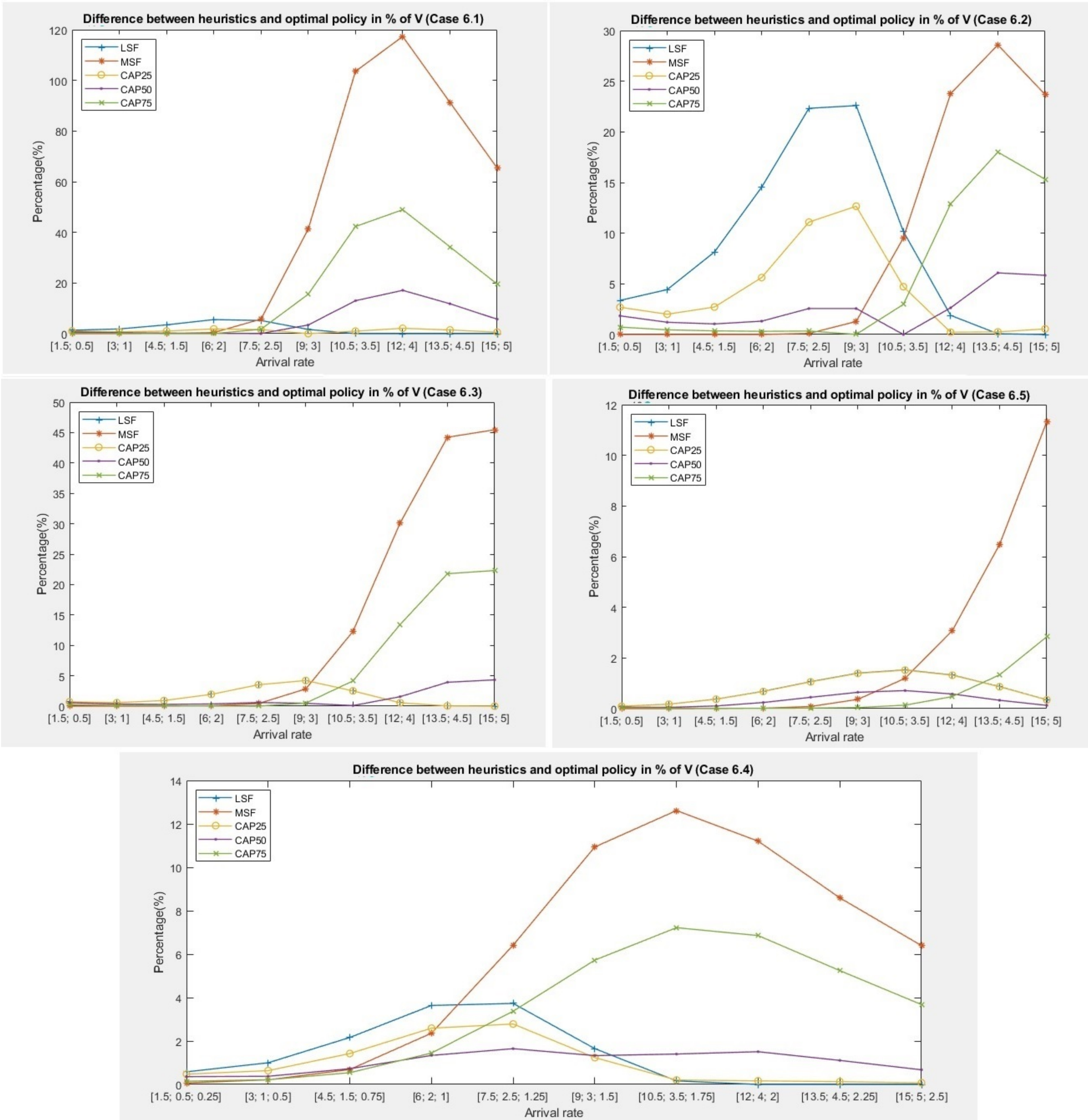


Figure 15: Comparison: Case 6.1, 6.2, 6.3, 6.4, 6.5

7 Discussion

7.1 Conclusion and implications

In this conclusion a summary of the most relevant results and their implications are presented. This study researched dynamic state based patient scheduling with a scarce healthcare server and beds in a finite priority queue. The problem, originally modeled as a CTMDP, was solved with the use of Matlab. Two different MDP models were analyzed, one without internal recovery and one with internal recovery.

Firstly, the results indicate that the optimal scheduling policy to save the most lives, under certain conditions, is a state based policy rather than a static policy. More specifically, a switching curve indicating when to switch from the most severe patient to the least severe was found in some of the scheduling policies. It does not come as a surprise that when more patients are waiting in the pre-admission room, it becomes more interesting to treat those patients that require less treatment time to prevent patients from being blocked. Changing some of the parameter values result in a different switching curve. A more interesting matter of changing parameter values is the change of arrival rates. The results indicate that the state-based dependent policy should change depending on the arrival rate. In practice, the arrival rate can be foreseen if there is knowledge of a massive accident or the contamination rate for a certain disease is known. The healthcare department can change its scheduling policy accordingly. For example, during corona if it is known that the contamination rate has increased, then there is a very high possibility that the arrival rate will increase as well and healthcare department could anticipate on this.

Secondly, the results of the multiple server case and internal recovery cases indicate that the state based policy should also depend on what kind of patients are currently in treatment and what kind of patients are currently in recovery. It makes sense that both should be taken into account when scheduling patients. If most servers are occupied with treating very severe patients that require a long treatment time, then this suggests that it might take a while before one of the occupied servers become available again. The opposite is true if most servers are occupied with treating less severe patients, then it is very likely that a server will be available again pretty soon. Almost the same reasoning can be applied to patients in recovery. As if most patients that are in recovery require a long recovery time, then this suggests that it might take a while for a bed to become available again. The server patient distribution is related to the odds of a server becoming available again and the recovery patient distribution is related to the odds of a bed becoming available again.

Thirdly, the comparison between simple heuristics and the optimal policies indicate that a simple rule of knowing when to switch from the most severe to the least severe can already lead to an improvement in saving more lives in comparison with just using one of the two.

Overall, this study's findings will contribute to the academic literature and practice in showing the importance of considering the patient mix waiting for treatment, but also in treatment and in recovery, when scheduling waiting patients. It further strengthens the statement of Xie et al. (2016), Ferrand et al. (2018) and Cao (2012) of taking into account the system state. As has been suggested in the introduction, this study was motivated by Covid-19 and the focus was on saving the most lives. Therefore, the results might be less interesting for the ED triage under normal conditions. However, it might still be interesting for future research under normal conditions, to take the aforementioned findings into account or for scheduling with QALY as objective function. Finally, this study provided a mathematical MDP model, which can be used as a basis for more extensive MDP models for healthcare systems.

7.2 Limitations

First of all, it should be noted that in reality healthcare is a very complex system and a lot of simplifications have been made. There are a lot of different diseases and people differ a lot in general and therefore not every disease has the same effect on each person. All variables, like the deterioration rate, probability of successful treatment, and mortality rate can differ heavily between patients. Those variables could also change during waiting depending on different circumstances. In this study, a maximum of three priority classes was defined, but in reality, it would be possible to divide all waiting patients into more priority classes. Moreover, the objective function of minimizing the number of deaths is also a simplification. As it is also reasonable to consider differences in eventual morbidity after treatment (e.g. not being able to walk the rest of your life). However, minimizing the number of deaths is a far more reasonable objective to consider, during massive incidents (pandemics) as corona.

Secondly, in this research, the experimental cases were limited by the processing power of the used notebook. Value iteration and policy iteration with an enormous state space can be a very big strain on the processor. As has been shown in the computation times, introducing an extra class of patients or priority or variable would heavily reduce the number of beds the notebook was able to analyze. Furthermore, multiple crashes and freezes of the used notebook, while testing several experimental cases, has lead us to become wary of trying out introducing more variables or priority classes.

Thirdly, in this research it was always assumed that the more severe a patient becomes the more treatment time it requires. One might argue that in reality, someone that is in a worse condition might not always require more treatment time then someone in a less severe condition, for example due to a younger age or a difference in injuries. However, in this research as we consider deterioration, it does not make sense that if a less severe patient deteriorates to a more severe state, that the required treatment time reduces.

7.3 Future research

The first possible direction to mention for future research is studying this problem with a very good processor. Realistically as technology keeps improving and supercomputers are introduced, this MDP problem can be solved for far bigger state spaces and thus more variables, like priorities or morbidity, can be introduced.

The second possible direction for future research is investigating this problem with different doctors, where the treatment time is not only based on the severity of the patient's state but also based on which doctor is going to treat the patient. In reality, the best doctor might be able to heavily reduce treatment time, but would it be the most optimal action to assign this doctor to the first patient in the queue or wait for a more severe patient to come in.

The third possible direction for future research is finding the switching curve. As the form of the scheduling policy is known as a switching curve, it might be interesting to approach the problem in a different manner to find this switching curve. Rather than fully developing an optimal action per state. However, if more variables are introduced, as recovery, then a simple switching curve might not suffice anymore.

References

Aloyce, R., Leshabari, S., & Brysiewicz, P. (2014). Assessment of knowledge and skills of triage amongst nurses working in the emergency centres in dar es salaam, tanzania. *African Journal of Emergency Medicine*, *4*(1), 14–18.

Argon, N. T., Ziya, S., & Righter, R. (2008). Scheduling impatient jobs in a clearing system with insights on patient triage in mass casualty incidents. *Probability in the Engineering and Informational Sciences*, *22*(3), 301.

Ashour, O. M., & Kremer, G. E. O. (2016). Dynamic patient grouping and prioritization: a new approach to emergency department flow improvement. *Health care management science*, *19*(2), 192–205.

Ashour, O. M., & Okudan, G. E. (2010). Fuzzy ahp and utility theory based patient sorting in emergency departments. *International Journal of Collaborative Enterprise*, *1*(3-4), 332–358.

Atar, R., Giat, C., & Shimkin, N. (2010). The cμ/θ rule for many-server queues with abandonment. *Operations Research*, *58*(5), 1427–1439.

Azadeh, A., Farahani, M. H., Torabzadeh, S., & Baghersad, M. (2014). Scheduling prioritized patients in emergency department laboratories. *Computer methods and programs in biomedicine*, *117*(2), 61–70.

Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., & Escobar, G. (2014). Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs*, *33*(7), 1123–1131.

Bellman, R. (1957). A markovian decision process. *Journal of mathematics and mechanics*, 679–684.

Beveridge, R., Clarke, B., Janes, L., Savage, N., Thompson, J., Dodd, G., … Vadeboncoeur, A. (1998). Implementation guidelines for the canadian emergency department triage & acuity scale (ctas). *Canadian association of emergency physicians*.

Cao, H., & Huang, S. (2012). Principles of scarce medical resource allocation in natural disaster relief: a simulation approach. *Medical Decision Making*, *32*(3), 470–476.

Christian, M. D. (2019). Triage. *Critical care clinics*, *35*(4), 575.

Cildoz, M., Ibarra, A., & Mallor, F. (2019). Accumulating priority queues versus pure priority queues for managing patients in emergency departments. *Operations Research for Health Care*, *23*, 100224.

Considine, J., LeVasseur, S. A., & Villanueva, E. (2004). The australasian triage scale: examining emergency department nurses' performance using computer and paper scenarios. *Annals of emergency medicine*, *44*(5), 516–523.

Derlet, R. W., Richards, J. R., & Kravitz, R. L. (2001). Frequent overcrowding in us emergency departments. *Academic Emergency Medicine*, *8*(2), 151–155.

Down, D. G., Koole, G., & Lewis, M. E. (2011). Dynamic control of a single-server system with abandonments. *Queueing Systems*, *67*(1), 63–90.

Fernandes, M., Vieira, S. M., Leite, F., Palos, C., Finkelstein, S., & Sousa, J. M. (2020). Clinical decision support systems for triage in the emergency department using intelligent systems: a review. *Artificial Intelligence in Medicine*, *102*, 101762.

Ferrand, Y. B., Magazine, M. J., Rao, U. S., & Glass, T. F. (2018). Managing responsiveness in the emergency department: Comparing dynamic priority queue with fast track.

Journal of Operations Management, *58*, 15–26.

Fitzgerald, A., & Wu, Y. (2017). Beyond clinical priority: what matters when making operational decisions about emergency surgical queues? *Australian Health Review*, *41*(4), 384–393.

Foster, E. M., Hosking, M. R., & Ziya, S. (2010). A spoonful of math helps the medicine go down: an illustration of how healthcare can benefit from mathematical modeling and analysis. *BMC medical research methodology*, *10*(1), 60.

Ghanbari, V., Ardalan, A., Zareiyan, A., Nejati, A., Hanfling, D., & Bagheri, A. (2019). Ethical prioritization of patients during disaster triage: A systematic review of current evidence. *International emergency nursing*, *43*, 126–132.

Harzi, M., Condotta, J.-F., Nouaouri, I., & Krichen, S. (2017). Scheduling patients in emergency department by considering material resources. *Procedia computer science*, *112*, 713–722.

He, Q.-M., Xie, J., & Zhao, X. (2012). Priority queue with customer upgrades. *Naval Research Logistics (NRL)*, *59*(5), 362–375.

Hinson, J. S., Martinez, D. A., Cabral, S., George, K., Whalen, M., Hansoti, B., & Levin, S. (2019). Triage performance in emergency medicine: a systematic review. *Annals of emergency medicine*, *74*(1), 140–152.

Howard, R. A. (1960). Dynamic programming and markov processes.

Ibe, O. (2013). *Markov processes for stochastic modeling.* Newnes.

Kleinrock, L. (1964). A delay dependent queue discipline. *Naval Research Logistics Quarterly*, *11*(3-4), 329–341.

Lakshmi, C., & Iyer, S. A. (2013). Application of queueing theory in health care: A literature review. *Operations research for health care*, *2*(1-2), 25–39.

Li, N., & Stanford, D. A. (2016). Multi-server accumulating priority queues with heterogeneous servers. *European Journal of Operational Research*, *252*(3), 866–878.

Lippman, S. A. (1975). Applying a new device in the optimization of exponential queuing systems. *Operations research*, *23*(4), 687–710.

McQuarrie, D. (1983). Hospitalization utilization levels. the application of queuing. theory to a controversial medical economic problem. *Minnesota medicine*, *66*(11), 679–686.

Meislin, H. W., Coates, S. A., Cyr, J., & Valenzuela, T. (1988). Fast track: Urgent care within a teaching hospital emergency department: Can it work? *Annals of emergency medicine*, *17*(5), 453–456.

O'Brien, D., Williams, A., Blondell, K., & Jelinek, G. A. (2006). Impact of streaming" fast track" emergency department patients. *Australian Health Review*, *30*(4), 525–532.

Pines, J. M., Hilton, J. A., Weber, E. J., Alkemade, A. J., Al Shabanah, H., Anderson, P. D., ... others (2011). International perspectives on emergency department crowding. *Academic Emergency Medicine*, *18*(12), 1358–1370.

Pollaris, G., & Sabbe, M. (2016). Reverse triage: more than just another method. *European Journal of Emergency Medicine*, *23*(4), 240–247.

Rahimi, S. A., Jamshidi, A., Ruiz, A., & Aït-Kadi, D. (2016). A new dynamic integrated framework for surgical patients' prioritization considering risks and uncertainties. *Decision Support Systems*, *88*, 112–120.

Reinhardt, M. R. (2017). A systematic approach to evaluation of performance deficiencies in ed triage. *Journal of Emergency Nursing*, *43*(4), 329–332.

Saghafian, S., Hopp, W. J., Van Oyen, M. P., Desmond, J. S., & Kronick, S. L. (2012). Patient streaming as a mechanism for improving responsiveness in emergency departments. *Operations Research*, *60*(5), 1080–1097.

Sanchez, M., Smally, A. J., Grant, R. J., & Jacobs, L. M. (2006). Effects of a fast-track area on emergency department performance. *The Journal of emergency medicine*, *31*(1), 117–120.

Sharif, A. B., Stanford, D. A., Taylor, P., & Ziedins, I. (2014). A multi-class multi-server accumulating priority queue with application to health care. *Operations Research for Health Care*, *3*(2), 73–79.

Smith, W. E., et al. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, *3*(1-2), 59–66.

Stanford, D. A., Taylor, P., & Ziedins, I. (2014). Waiting time distributions in the accumulating priority queue. *Queueing Systems*, *77*(3), 297–330.

Van Mieghem, J. A. (1995). Dynamic scheduling with convex delay costs: The generalized c— mu rule. *The Annals of Applied Probability*, 809–833.

Wuerz, R. C., Milne, L. W., Eitel, D. R., Travers, D., & Gilboy, N. (2000). Reliability and validity of a new five-level triage instrument. *Academic emergency medicine*, *7*(3), 236–242.

Xie, J., Cao, P., Huang, B., & Ong, M. E. H. (2016). Determining the conditions for reverse triage in emergency medical services using queuing theory. *International Journal of Production Research*, *54*(11), 3347–3364.

Xie, J., He, Q.-M., & Zhao, X. (2008). Stability of a priority queueing system with customer transfers. *Operations Research Letters*, *36*(6), 705–709.

Zachariasse, J. M., van der Hagen, V., Seiger, N., Mackway-Jones, K., van Veen, M., & Moll, H. A. (2019). Performance of triage systems in emergency care: a systematic review and meta-analysis. *BMJ open*, *9*(5), e026471.

8 Appendices

8.1 Appendix A: Literature table

Article	General	Multiple servers	Patient's condition dependent priority allocation	Patient's condition dependent result treatment	Abandonment patient or customer or job	Deterioration patient/priority change customer	Scope	Method
Sanchez, M., Smally, A. J., Grant, R. J., & Jacobs, L. M. (2006). Effects of a fast-track area on emergency department performance. The Journal of emergency medicine, 31*(1), 117-120.*	Comparison of FT with most serious first. FT can lead to a decrease in average length of stay (LOS) and percentage of patients leaving without being seen(LWBS), without a harmful effect on the revisit rate and mortality rate.	Multiple real doctors	Emergent, urgent or non-urgent	N/A	N/A	N/A	Real life analysis in a hospital	Data evaluation
Argon, N. T., Ziya, S., & Righter, R. (2008). Scheduling impatient jobs in a clearing system with insights on patient triage in mass casualty incidents. Probability in the Engineering and Informational Sciences, 22*(3), 301.*	Provide multiple non-state dependent and state dependent heuristics to minimize total number of abandonments. State-dependent heuristics perform better than non-state dependent heuristics in moments of high congestion.	N/A	Two priority classes of patients and heuristics decide who is treated first	N/A	Both priority classes of patients can abandon the system, with different exponential distributed rates.	N/A	Healthcare triage	Heuristic comparison against dynamic programming
Sharif, A. B., Stanford, D. A., Taylor, P., & Ziedins, I. (2014). A multi-class multi-server accumulating priority queue with application to health care. Operations Research for Health Care*, 3(2), 73-79.*	Discusses a multi-server accumulation priority queue model for an arbitrary number of customer classes.	Two servers	Patients are initially triaged with CTAS, but their priority increases over time	N/A	N/A	N/A	Emergency Department	Analytically and simulation approach
Xie, J., Cao, P., Huang, B., & Ong, M. E. H. (2016). Determining the conditions for reverse triage in emergency medical services using queuing theory. International Journal of Production Research, 54*(11), 3347-3364.*	They investigate the conditions for prioritizing a lower urgency patient over a higher urgency patient. They develop a queueing model with transfers and abandonments.	N/A	Two priority classes of patients. Allocation depends on optimal policy.	N/A	High priority patient can leave the system after random exponential distributed time.	Yes, low priority patient can change into a high priority patient after random exponential distributed time.	Emergency Department	Infinite Continuous Markov Decision process, solved with smoothed rate truncation method and value iteration.
Cildoz Esquíroz, M., Ibarra, A., & Mallor Giménez, F. (2019). Accumulating priority queues versus pure priority queues for managing patients in emergency departments. Operations Research For Health Care, 23 (2019) 100224.	Discusses a multi-server accumulation priority queue with a finite horizon model.	Multiple servers	Patients are initially triaged with CTAS, but their priority increases over time	N/A	N/A	N/A	Emergency Department	Discrete Event Simulation
Azadeh, A., Farahani, M. H., Torabzadeh, S., & Baghersad, M. (2014). Scheduling prioritized patients in emergency department laboratories. Computer methods and programs in biomedicine, 117*(2), 61-70.*	The patient scheduling problem was formulated as a generalized flexible open shop problem and a MILP model was proposed. A genetic algorithm was developed for solving the problem	Multiple different servers	Healthcare pathway dependent on priority patients	N/A	N/A	N/A	Emergency Laboratories	MILP model with Genetic algorithm
This research	Optimal scheduling policies are developed for scheduling deteriorating patients with one and multiple servers in a finite queue. Extension is provided with recovery in healthcare department and early discharge possibilities.	One and multiple servers	Severity of patient's condition determine the classification. Scheduling based on optimal policy.	Probability on successful treatment	Highest priority patient can leave the healthcare department after an exponential distributed time.	Yes, priority of patient can increase after exponential distributed time.	Healthcare department	Finite Continuous Markov Decision Process, solved with uniformization and policy iteration.

Figure 16: Most important characteristics of literature review

8.2 Appendix B: Discount factor analysis

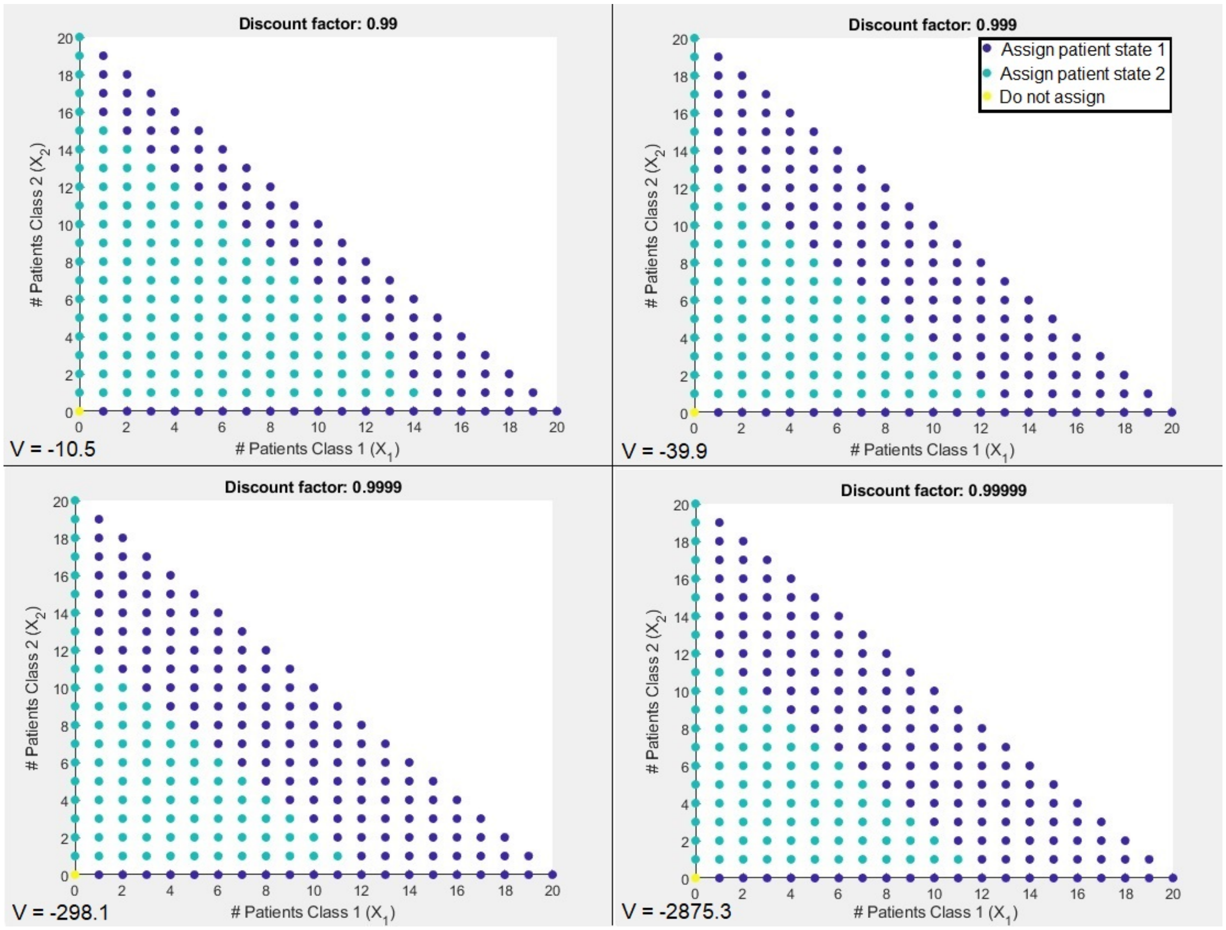


Figure 17: Discount factor analysis cases

8.3 Appendix C: Case 2.2 and case 2.4

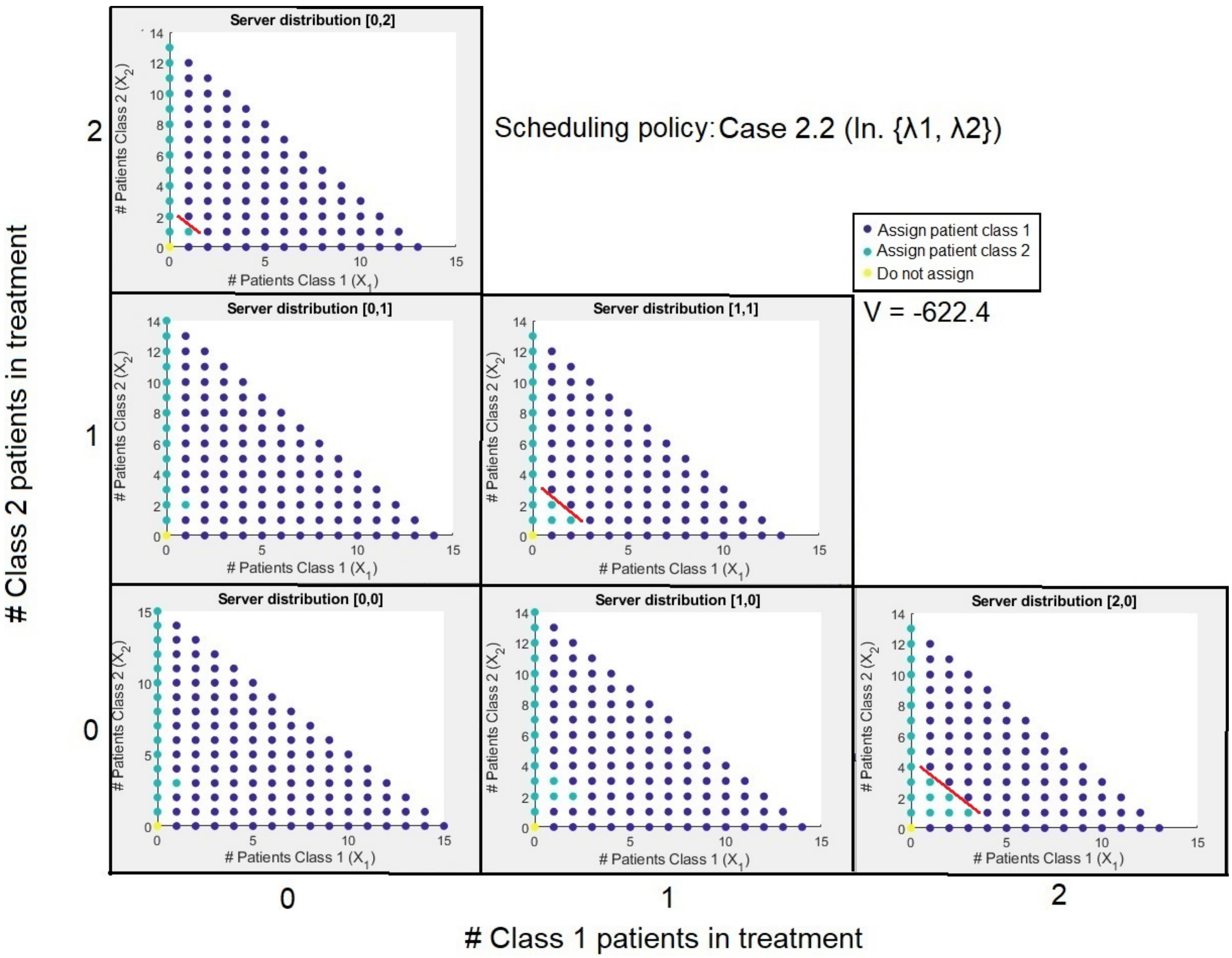


Figure 18: Case 2.2

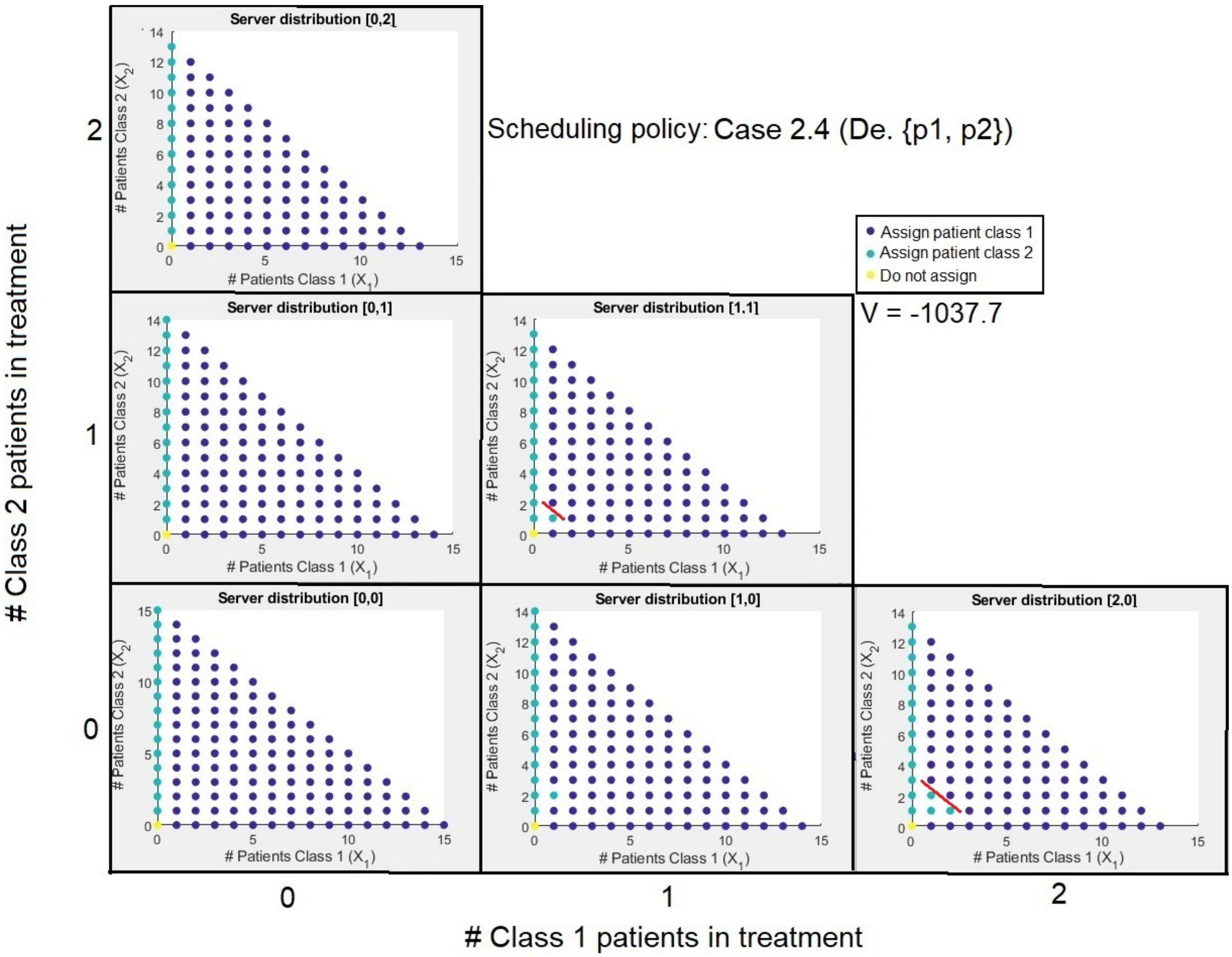


Figure 19: Case 2.4

8.4 Appendix D: Recovery distribution tables

Treat class 1 patient if			
X_1	X_2	Z_1	Z_2
2	3	0	5
3	2	0	4
		0	5
		1	4
4	1	0	3
		0	4
		0	5
		1	3
		1	4
		2	2
		2	3
		3	2

Table 16: Recovery table case 5.1

Treat class 1 patient if			
X_1	X_2	Z_1	Z_2
1	6	0	3
		1	2
2	5	0	2
		0	3
		1	2
		2	1
2	4	0	4
3	1	0	6
3	2	0	5
3	3	0	4
		1	3
3	4	0	2
		0	3
		1	1
		1	2
		2	1
		3	0
4	1	0	5
		1	4
		2	3
4	2	0	3
		0	4
		1	3
		2	2
5	1	0	3
		0	4
		1	2
		1	3
		2	2
		3	1

Treat class 2 patient if			
X_1	X_2	Z_1	Z_2
4	3	1	0
		0	0
5	2	0	0

Table 17: Recovery table case 5.2