Heuristics for inventory management of reusable medical device trays

Bachelor Thesis Advanced Technology

Wick Wijnholds

August 2023

Supervisor:
dr. ir. Aleida Braaksma

Committee:
dr. ir. Aleida Braaksma
Hayo Bos, MSc
dr. Faiza Bukhsh

CHOIR Diakonessenhuis

UNIVERSITY OF TWENTE.

Abstract

This report summarizes our research into heuristics for inventory levels of reusable medical device (RMD) trays in hospitals, with a specific focus on the case of the Diakonessenhuis, a medium-sized hospital in The Netherlands. We show that generic methods from the field of inventory management are not applicable to the case of RMD tray inventories, and describe and compare domain-specific heuristics described by Fineman and Kapadia [1], Diamant et al. [2] and Hosteins et al. [3], using a discrete event simulation by the latter. We show that the heuristic by Diamant et al. performs best, provided suitable parameters are chosen. Finally, we look into the potential of operating theatre planning modification and the pooling of intercompatible tray types, and find that there is potential in the former, but fail to show potential in the latter.

CONTENTS

1 Introduction

1.1 Background

In recent years, hospitals have had to deal with a steady increase in patient and procedure numbers [4]. As a result, healthcare costs have steadily been rising. Surgical procedures, performed in operating rooms (ORs), account for up to 60% of total expenses, of which about half is spent on equipment [5]. The inventory of surgical instruments alone is often worth several millions of dollars [6]. It is therefore critical to manage this costly inventory properly. In this report, we summarise our research into heuristics for reusable medical device (RMD) inventory levels in hospitals - from literature study to a comparison of selected heuristics using discrete-event simulation, based on data from a medium-sized hospital in The Netherlands.

1.2 RMD usage patterns

The usage of RMDs involves a cyclic pattern: they are first used in an OR, then transported to a central sterilisation service (CCS), where they go through several stages of cleaning and sterilisation. Typically, this process takes several hours. They are then moved to inventory, ready to be used again. RMDs are not generally packaged separately: instead, they are stocked in sets of varying sizes in surgical trays. Once opened, these trays need to be fully sterilised before they can be re-used. There is often a multitude of different trays, each tailored for specific surgical procedures. Managing all this inventory is complicated by the fact that the trays required vary not only per procedure, but also per surgeon. We will explore the usage of RMDs further in Section 2.

1.3 Problem and Approach

The goal of our research is to propose a heuristic for RMD inventory levels (based on existing inventory management best practices, in combination with the outcomes from simulations) that allows the hospital to better tailor inventory levels to demand, either by reducing total inventory or reducing the rate at which the trays are not available when they are needed.

Based on this goal, we formulated our research question:
“Can we find or formulate a heuristic for RMD tray inventory levels that, once implemented, performs better than the current historically determined inventory levels?”

The approach taken to answer this question is as follows:

1. Understand and describe the way that RMD trays are (re)used in hospitals.
2. Perform a literature search for techniques from inventory theory involving reuse, as well as for domain-specific literature (i.e. healthcare/hospital/sterilisation logistics).
3. Evaluate the results of the literature study, and choose heuristics to test.
4. Implement the chosen heuristics, and test them using the simulation by Hosteins et al. [3]
5. Process and evaluate the results of the above testing. If needed, alter the heuristics to improve performance.
6. Repeat the implementation–testing–processing–evaluation cycle until satisfied.

1.4 Report structure

This report is structured in the following way:
Section 2 gives a more thorough description of the problem. Section 3 then gives an overview of the results of the literature study. Section 4 then details the heuristics we selected. Section 5 describes our testing methodology. The results of our tests can be found in Section 6. Section 7 explores several potential avenues of improvement. Finally, Section 8 provides concluding remarks and discusses possible future research avenues, and Section 9 looks back at the research process.

2 Problem formulation

Let us now take a detailed look at the problem at hand, starting with the cycle that RMD trays go through, continuing on with a description of our case study, and ending with a description of the demand patterns of RMD trays.

2.1 RMD usage

After the RMD trays arrive at the CSS, the following steps are taken (see Figure 1):
First, the RMDs are taken out of the trays. The RMDs are then individually rinsed. The RMDs then go through a cleaning cycle in a disinfection machine. The RMDs are checked for damage and assembled into trays. Finally, they go through a sterilisation cycle in an autoclave. When they are ready for use, they go into storage, until such a time arrives that they are needed in the OR or in an outpatient facility. If the trays are available when needed, then all is well. If they are not available, hospital staff will

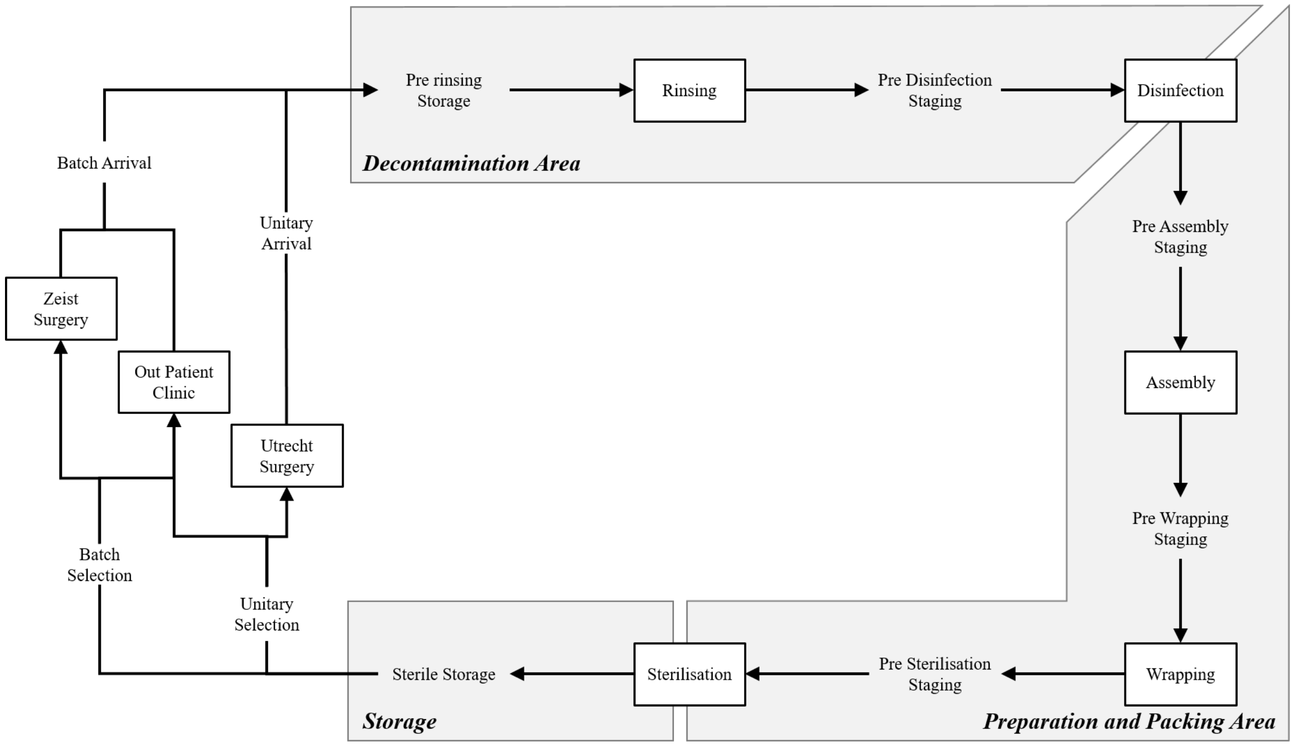


Figure 1: Sterilisation flow, from [3]

see if another set of trays could be used for the procedure at hand. If this is not the case, the surgery might be delayed slightly if the tray is almost sterile, or, more commonly, it is rescheduled to a later date if possible. This, of course, is not to be preferred, and happens quite infrequently. After usage, the trays return to the CSS, completing the cycle.

2.2 Case study

The Diakonessenhuis is a medium-sized hospital in the province of Utrecht, The Netherlands. It consists of one main location with a central sterilisation service (CSS) in the city of Utrecht and several subsidiary outpatient clinics in the greater Utrecht area. For these subsidiary locations, RMD trays are transported in batches to and from the CSS. Currently, RMD tray inventory levels are not determined in a consistent manner: they are, instead, mostly historically determined. This, of course, often leaves room for improvement, as it does in this case. In order to work more efficiently, the Diakonessenhuis has entered into a collaboration with the Centre for Healthcare Operations Improvement and Research (CHOIR) at the University of Twente. One of the results of this collaboration is a discrete event simulation of the sterilisation process by Hosteins et al. [3], which was tailored to the situation at the Diakonessenhuis and verified against data collected at the hospital.

2.3 Tray demand

As part of their research, Hosteins et al. [3] analysed the usage patterns of RMD trays. They note that there are multiple sources of demand:

- Elective surgeries, which are planned ahead of time and might require multiple trays.
- Emergency surgeries, which are obviously not planned ahead of time, and might require multiple trays.
- Outpatient procedures, which are planned ahead of time and require only one tray. These account for about ten percent of surgeries.

In order to ease analysis of the results for the 1213 tray types present at Diakonessenhuis, Hosteins et al. [3] divided these types into three clusters:

- Cluster 0 comprises the 32 most frequently used trays in surgeries - these have high demand and appropriately high inventory levels.
- Cluster 1 comprises the 771 other trays used in surgeries.
- Cluster 2 comprises the 410 trays used in outpatient procedures.

We will use the same clusters throughout this report.

2.3.1 The trouble with demand

Many of the methods that we will explore will require us to put a number on the demand for a specific tray type. This is not as easy as it may seem - not only does the level of demand vary wildly between tray types, it also varies significantly from day to day. There are even a few tray types that have zero demand altogether. Compare, for example, the demand for tray 0 (in Figure 2) and tray 4 (in Figure 3) - tray 0 has relatively stable and high demand outside of the weekend, while tray 4 has low demand, and only on Wednesdays and Thursdays.

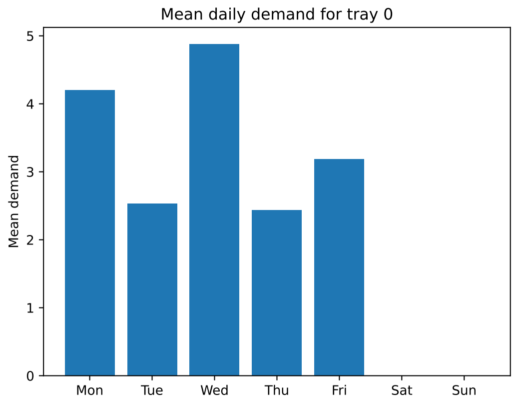


Figure 2: Daily mean demand for trays of type 0 (Cluster 0)

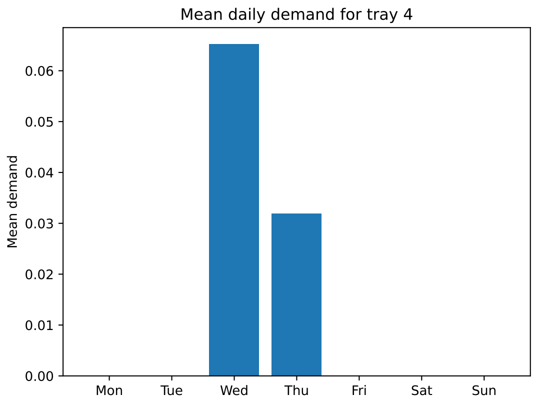


Figure 3: Daily mean demand for trays of type 4 (Cluster 2)

We can see that this is exemplary of a difference in clusters when we look at Figures 4 and 5: high demand in cluster 0, low demand in clusters 1 and 2, and a much greater variability of demand in cluster 2 than in cluster 0, with cluster 1 in between.

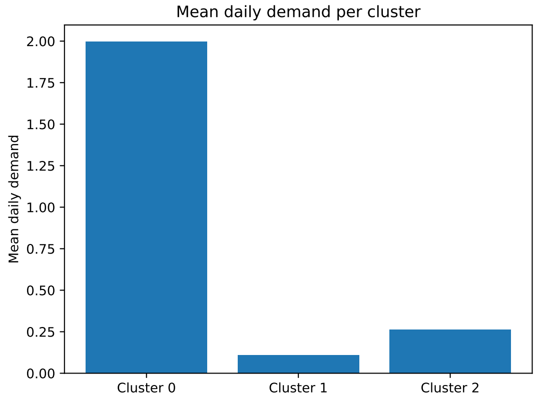


Figure 4: Mean demand over all trays of each cluster

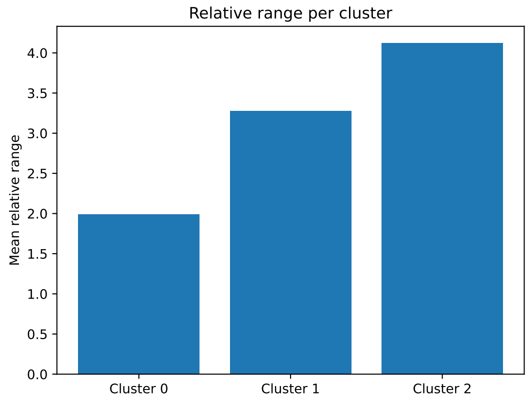


Figure 5: Relative range (difference between maximum and minimum, divided by the mean), averaged over all trays of each cluster

If we are now asked to pick a single number to characterise 'demand', as we often will be, what would that number be? In this report, we often choose the demand on the busiest day of the week. We believe we can justify this, because the worst-case performance is very relevant to us. After all, the hospital will never run out of trays of type 4 on a Monday, but it might do so on a Wednesday. When we must specify what distribution the demand has, we will work under the assumption of Poisson arrivals for all procedures, which by the additive property of the Poisson distribution also implies that the demand for each tray type will be Poisson-distributed. This is the same assumption as made in the simulation by Hosteins et al. [3]. De Bruin et al. [7] also found this to be true for admissions to hospital wards in another Dutch hospital, which we expect translates well to surgery demand.

3 Literature review

3.1 Inventory management and reverse logistics

Because of the cyclic nature of the use of RMD trays, we looked into the field of 'closed-loop supply chain' (CLSC) inventory theory, which is also known as 'reverse logistics' (RL). The methods used in this field are often adapted from those used in classical inventory theory to account for the presence of a loop in what would otherwise be a straight supply chain. Of course, there is a multitude of ways in which a supply chain could contain such a closed loop - a classification by Akçali and Çetinkaya categorised research until 2009 by configuration, and found 14 different closed-loop supply chain types [8]. When we look at our case through their classification system, we find that it can best be described as a so-called 2SP-a system: a system with two stock points (in our case, pre-sterilisation and post-sterilisation storage), without manufacturing (see Figure 6).

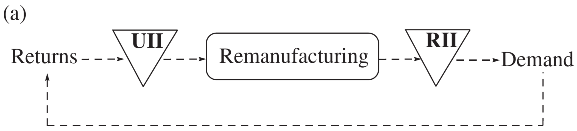


Figure 6: The 2SP-a system, courtesy of [8]

Many of the more complicated configurations reduce to 2SP-a, expanding the amount of research that we can make use of significantly. Take, for example, a configuration where products can be made from either new materials or recycled materials. When we set the price of new materials to a prohibitively high level, we can force any applicable heuristics to produce a minimal number of products from scratch. We have then, effectively, reduced the configuration of the system to one which is closer to our system. In this way, we can at least attempt to apply a decent fraction of research to our problem. To do this, we must translate our problem into a reverse logistics problem. At first glance, this appears to be merely a case of semantics: where we *sterilise* an RMD tray, they *remanufacture* goods. Where we *purchase* RMD trays, they *manufacture* goods. Where they get *returns* or store *recoverables*, we have *dirty trays*. When they talk about *servicables*, we talk about *sterile inventory*. When we dig deeper, however, the divide turns out to be broader than terminology. Let us look at some of the approaches from reverse logistics, and see where the analogy breaks down.

3.1.1 EOQ-based

The economic order quantity (EOQ) is a concept in standard inventory management, providing an exact answer to the question: 'what is the optimal size of orders such that the total inventory cost is minimised?'. It applies specifically to situations of constant demand and instantaneous production, which simplifies its calculation. In their book on reverse logistics, Dekker et al. [9] suggest several different modifications of the EOQ model for reverse logistics, where some fraction of demand is met with remanufactured goods, while other goods are produced from scratch. They optimise the total cost per unit of time, i.e., the sum of production and remanufacturing setup costs and holding costs for servicables and recoverables under various (re)manufacturing schedules. Of these, the most applicable is a modification where it is assumed one manufacturing batch is followed by R remanufacturing batches. Dekker et al. find the following formula for an optimal initial manufacturing batch size Q_p^*:

$$Q_p^* = \sqrt{\frac{2(d-u_r)K_p}{h_s(1-\frac{u_r}{d}) + h_u\frac{u_r}{d}}} \tag{1}$$

where d is the demand, u_r is the number of returns, K_p is the manufacturing setup cost, h_s is the unit holding cost of servicables and h_u is the unit holding cost of recoverables. This formula, however, starts to break down when $\frac{u_r}{d}$ approaches 1, i.e., when the return rate for a given time frame is very high, when it reduces to the following:

$$Q_p^* \approx \sqrt{\frac{2(d-u_r)K_p}{h_u}} \tag{2}$$

We find that Q_p^* is now determined by K_p, h_u and $d - u_r$. When we translate this back to our case, we see that the optimal inventory size (that is what Q_p^* would be) is determined by:

- Manufacturing setup costs, which become the fixed costs incurred when ordering RMD trays (which will probably be dominated by shipping costs, relatively minor in comparison to the cost of the trays themselves).
- The cost of storing the dirty trays (which is zero for all intents and purposes, since trays are sterilised almost immediately after use, so they spend only a short time on a small shelf waiting for their turn). Given that $h_u \approx 0$, the value of Q_p^* will be unrealistically high.
- What might best be described as of 'net demand' $(d - u_r)$, which approaches zero as $\frac{u_r}{d} \to 1$.

All the above makes clear that while this heuristic might work excellently for, say, a beer bottle manufacturer (with high manufacturing setup costs for firing up the glass furnaces, lower reuse rates and significant costs for storage), it does not translate well to our problem.

3.1.2 Wagner-Whitin-based

In 1958, Harvey Wagner and Thomas Whitin [10] developed an algorithm to solve the so-called dynamic lot size model: to find a production schedule that minimises total inventory costs for variable (forecast) demand over a set number of time slots. They formulated a mixed-integer linear programming (ILP) problem to do just that. In 2000, Richter and Sombrutzki modified this model to take into account remanufacturing [9, 11] and set up the following ILP:

$$
\begin{aligned}
\min C = \sum_{t=1}^{T} (K_p\gamma_t^p + K_r\gamma_t^r + K_w\gamma_t^w + c_pQ_t^p \\
+ c_rQ_t^r + c_wQ_t^w + h_sI_t^o + h_uI_t^u) \\
\text{s.t. } I_t^o = I_{t-1}^o - d_t + Q_t^p + Q_t^r \\
I_t^u = I_{t-1}^u + u_t - Q_t^r - Q_t^w \\
Q_t^p \leq M\gamma_t^p, Q_t^r \leq M\gamma_t^r, Q_t^w \leq M\gamma_t^w \\
Q_t^p, Q_t^r, Q_t^w, I_t^0, I_t^u \geq 0 \\
\gamma_t^p, \gamma_t^r, \gamma_t^w \in \{0, 1\},
\end{aligned}
$$

where:

- K_p, K_r and K_w are the fixed setup costs for production, remanufacturing and disposal, respectively.
- γ_t^p, γ_t^r and γ_t^w indicate whether production, remanufacturing or disposal occur in period t.
- c_p, c_r and c_w are the variable costs per unit for production, remanufacturing and disposal, respectively.
- Q_t^p, Q_t^r and Q_t^w is the number of units produced, remanufactured or disposed of in period t.
- h_s and h_u are holding costs for servicables and recoverables, respectively.
- I_t^o and I_t^u are the inventory levels for servicables and recoverables in period t, respectively.
- d_t and u_t are the forecast demand and returns in period t, respectively.

When we analyse this ILP in relation to our problem, we note the following:

1. Disposal is not relevant to our case - we are not throwing away perfectly good RMDs. That also means that we will always clean them - the variable costs for remanufacturing do not matter either.
2. The CSS has fixed, long, continuous 'production runs' (i.e., working days of the CSS staff). There are no significant costs associated with switching from cleaning one type of tray to another, either. Remanufacturing setup costs, therefore, are not very relevant.
3. Similar to what we noted with the modified EOQ heuristic, 'production' setup costs are negligible, as are holding costs for recoverables.
4. Since remanufacturing is no longer a choice, it is predetermined, so we can remove the variable Q_t^r. We can instead say that the inventory remanufactured equals the demand from a prior period. In fact, we can stop keeping track of the recoverables inventory altogether.
5. We want to find an initial inventory level, so we do not need to purchase inventory after the first period.
6. After all these changes, the holding costs (which we now calculate only for sterile inventory) will scale very well with the initial inventory level. We can therefore just minimise the initial inventory level.

Removing these parts from the Richter-Sombrutzki ILP, we are left with the following simple scheme:

$$
\begin{aligned}
&\min I_0^o \\
&\text{s.t. } I_t^o = I_{t-1}^o - d_t + u_{t-1} \\
&Q_t^p, I_t^o \in \mathbb{N}
\end{aligned}
$$

where the demand d_t and return rate u_t are forecasts.

This ILP can be described as: "find the minimal inventory level such that demand from our forecast is met for all periods over which we give a forecast." This is not a very nuanced statement - it does not allow, for example, weighing some small fraction of RMD tray unavailability against the benefit of needing less inventory. It is more akin to the BSxx heuristic that we will discuss in Section 4.1, while being less versatile. For this reason, we must also conclude that it is probably not very useful.

Again, we see that our problem simply does not map very well onto techniques for production and order planning, which leads us to disregard techniques that are used throughout many industries. We therefore turn to domain-specific literature.

3.2 Domain-specific literature

More specific approaches can be found in the work of Bijvank and Vis [12], Diamant et al. [2] and Fineman and Kapadia [1].

3.2.1 Fineman and Kapadia

The earliest paper that we could identify relating to inventory levels of sterile inventory comes from Fineman and Kapadia [1], who, in 1978, developed a simple model for the logistics of sterilised items in hospitals. They made various simplifying assumptions, the most important of which are continuous processing and non-varying daily demand, and came up with heuristics for what they call 'replacement stock' - inventory to be ordered to replace worn out items - and 'processing stock' - inventory required to keep supply up while other inventory is processed, or in our case sterilised:

$$\text{Replacement Stock} = D(t_a + t_c)/n \qquad (3)$$

$$\text{Processing Stock} = D \cdot t_{\text{process}} \qquad (4)$$

where

$$\begin{aligned} D &= \text{the daily demand for the item in question} \\ t_a &= \text{the replenishment interval} \\ t_c &= \text{the time the hospital could be cut off} \\ &\quad \text{from supplies} \\ t_{\text{process}} &= \text{the processing time for this item} \\ n &= \text{the number of possible uses} \\ &\quad \text{of the item in question} \end{aligned}$$

For our case, where n is usually high (many RMDs can be used dozens of times), replacements do not take long to arrive (so t_a is relatively low), and supply chains are robust (so t_c is relatively low), the replacement stock levels will be mostly insignificant when compared to the processing stock levels. A simple heuristic might be to keep stocks equivalent to the processing stock levels proposed by Fineman and Kapadia, where we take the processing time to be the median time for an RMD tray to go through sterilisation, which is 3 hours and 18 minutes at the Diakonessenhuis.

3.2.2 Bijvank and Vis

Bijvank and Vis consider inventory processes in hospitals, and model these as discrete-time Markov chains [12]. Unfortunately, their work is mostly concerned with disposable inventory that needs constant replenishment, like gloves and needles. What is interesting in their work, however, is that they model the inventory of items through time as a discrete-time Markov chain - an approached shared with the research of Diamant et al., which we will now discuss.

3.2.3 Diamant et al.

Like Fineman and Kapadia, Diamant et al. [2] are actually concerned with our specific problem - that of reusable surgical supplies. As stated before, Diamant et al. use a discrete-time Markov chain for their inventory process. This is a Markov chain with states $y_t^S \in \{0, 1, ..., S-1, S\}$ where y_t^S is the amount of ready-to-use inventory in period t and S is the total number of RMD trays of a specific type. Every transition from one state to the next corresponds to the passing of a set amount of time. Transition probabilities are determined by demand, which can be arbitrarily distributed. In this model, the assumption is made that inventory used in one period becomes unavailable the next period, and available the period after that. Choosing the correct period length turns out to be very important, as we will see in Section 6. This heuristic is worked out in much more detail in Section 4.3.

3.3 Computer simulation approaches

Besides this work on heuristics, progress has also been made in a related field: building computer simulations of the sterilisation process. For example, in 2012, Di Mascolo and Gouin [13] created a generic model of sterilisation departments which uses data on the specific set-up of a sterilisation department in order to predict performance. More recent work by Rupnik, Narding and Kramberger in 2019 [14] and Hosteins, Bos and Leeftink in 2023 [3] uses discrete event simulation (DES) in combination with actual RMD utilisation data in order to simulate with higher accuracy and fewer simplifications.
The simulation by Rupnik et al. models the sterilisation department as a network of two queues (one after the other), each with three servers, and uses real data from 2000 surgeries for validation. The simulation by Hosteins et al. models the sterilisation department in much more detail, separating the process into eight diferent steps (see Figure 1), each with their own queue, and proposing an OR schedule generator in order to simulate actual usage patterns, thereby fully simulating the usage cycle of RMDs. We will use their simulation in order to assess our heuristics.

4 Heuristics

4.1 Base-stock heuristic

In their paper describing their DES model of the sterilisation cycle, Hosteins et al. [3] also tested three instances of their so-called 'base-stock' (BSxx) heuristic: BS50, BS75 and BS95. In this heuristic, the inventory level is set to the xxth percentile of demand on the busiest day of the week. We will use this heuristic as a point of comparison.

4.2 Fineman and Kapadia

As mentioned before in Section 3.2.1, we can use the processing stock heuristic from the paper of Fineman and Kapadia [1], which is as follows:

$$\text{Processing Stock} = D \cdot t_{process}$$

where

- D is the daily demand for the item in question
- $t_{process}$ the processing time for this item

Given that daily demand is not constant, we have several options to choose from. We tested the demand on the busiest, mean and median day of the week, based on the data provided with the model by Hosteins et al., all with a processing time of four hours.

4.3 Diamant et al.

As mentioned before, the heuristic by Diamant et al. uses a Markov chain to model the system. We now describe how this heuristic works. First, define the following variables:

- S: the number of RMD trays in the system
- y_t^S: The on-hand inventory at the start of period t, given S total trays
- d_t: demand in period t
- z_t^S: the number of trays to be sterilised at the end of period t given S total trays

Using the definition $(x)^+ = \max(x, 0)$, we then assume that any trays used in period t are sterilised in period $t+1$ and ready to be used again at the start of period $t+2$. Therefore, the on-hand inventory at the start of a new period equals the sum of the on-hand inventory at the end of the previous period and the inventory that was newly sterilised:

$$y_t^S = (y_{t-1}^S - d_{t-1})^+ \text{ for } t = 1$$
$$y_t^S = (y_{t-1}^S - d_{t-1})^+ + z_{t-2}^S \text{ for } t = 2, 3, ...$$

Since we can never use more inventory than we have on hand, the 'dirty' inventory at the end of the period is $z_t = \min(y_t^S, d_t)$. Important to note is that the total inventory is constant, which implies $S = y_t^S + z_{t-1}^S$: inventory is either available or being cleaned. We can use this to find a relationship between y_t^S, y_{t-1}^S and d_{t-1}:

$$y_t^S = (y_{t-1}^S - d_{t-1})^+ + S - y_{t-1}^S \tag{5}$$
$$= \begin{cases} S - y_{t-1}^S & \text{if } d_{t-1} \geq y_{t-1}^S \\ S - d_{t-1} & \text{if } d_{t-1} < y_{t-1}^S \end{cases} \tag{6}$$

Clearly, what we have here is a discrete-time Markov chain $\{y_t^S, t = 1, 2, ..., T\}$ where the transition probabilities P_{ij} from $y_{t-1}^S = i$ to $y_t^S = j$ are set by the distribution of the demand. Let $\{\pi_i(S) = \mathbb{P}(y_t^S = i)\}$ be the steady-state probability distribution of the inventory level. Assuming i.i.d. demand represented by a non-negative integer-valued random variable D, we can define the service level as the probability that demand does not exceed inventory:

$$\text{Service Level} = 1 - \sum_{i=0}^{S} \pi_i(S)\mathbb{P}(D > i) \tag{7}$$

Taking this as their starting point, Diamant et al. [2] find a closed-form solution for $\pi_0(S)$ and $\pi_S(S)$:

$$\pi_0(S) = \frac{\mathbb{P}(D = 0)\mathbb{P}(D \geq S)}{1 - \mathbb{P}(D \geq S)\mathbb{P}(D \geq 1)} \tag{8}$$
$$\pi_S(S) = \frac{\mathbb{P}(D = 0)}{1 - \mathbb{P}(D \geq S)\mathbb{P}(D \geq 1)} \tag{9}$$

Assuming Poisson demand with mean λ, we then find the following:

$$\begin{aligned} \mathbb{P}(D = 0) &= \frac{\lambda^0 e^{-\lambda}}{0!} = e^{-\lambda} \\ \mathbb{P}(D \geq S) &= 1 - \mathbb{P}(D \leq S - 1) \\ &= 1 - e^{-\lambda} \sum_{i=0}^{S-1} \frac{\lambda^i}{i!} \\ \mathbb{P}(D \geq 1) &= 1 - \mathbb{P}(D = 0) \\ &= 1 - e^{-\lambda} \end{aligned}$$

which gives us $\pi_0(S)$ and $\pi_S(S)$:

$$\pi_0(S) = e^{-\lambda} \frac{1 - e^{-\lambda} \sum_{i=0}^{S-1} \frac{\lambda^i}{i!}}{1 - (1 - e^{-\lambda} \sum_{i=0}^{S-1} \frac{\lambda^i}{i!})(1 - e^{-\lambda})} \tag{10}$$
$$\pi_S(S) = e^{-\lambda} \frac{1}{1 - (1 - e^{-\lambda} \sum_{i=0}^{S-1} \frac{\lambda^i}{i!})(1 - e^{-\lambda})} \tag{11}$$

These are both monotone decreasing functions of S, as can be seen in Figures 7 and 8 and is proven by Diamant et al [2].

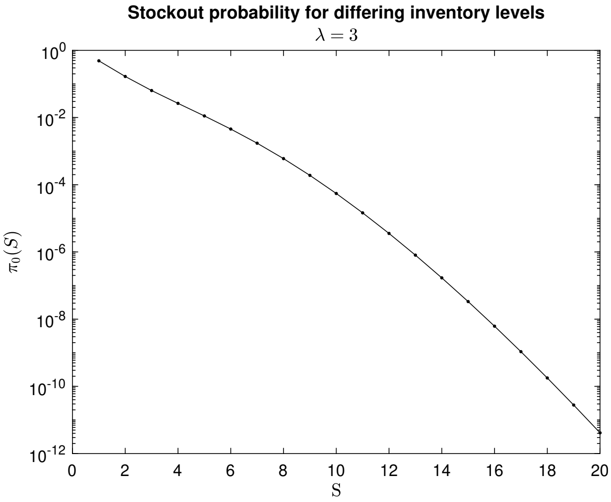


Figure 7: Stockout probabilities for varying inventory levels

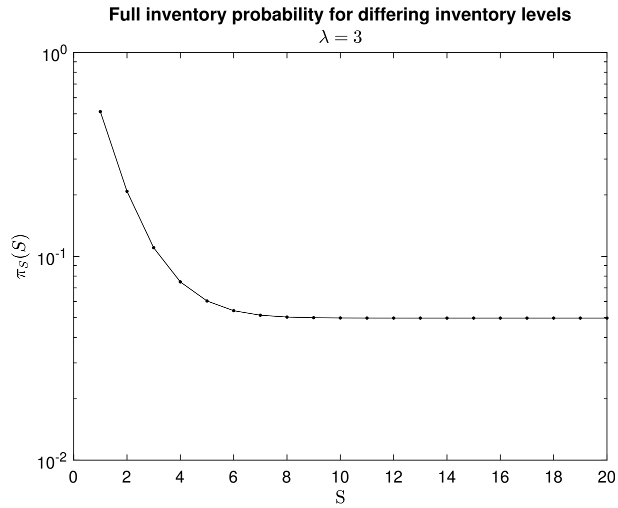


Figure 8: Full inventory probabilities for varying inventory levels

The rest of the steady-state probability distribution can then be solved for using a recursive method [2]:

$$\pi_{S-i}(S) = \frac{\mathbb{P}(D = i)}{1 - \mathbb{P}(D \geq S - i)\mathbb{P}(D \geq i + 1)} \times \left(1 - \sum_{k=0}^{i-1} \pi_k(S)\right) + \frac{\mathbb{P}(D \geq i + 1)\mathbb{P}(D = S - i)}{1 - \mathbb{P}(D \geq S - i)\mathbb{P}(D \geq i + 1)} \times \sum_{k=S+1-i}^{S} \pi_k(S)$$

$$\pi_i(S) = \frac{\mathbb{P}(D \geq S - i)\mathbb{P}(D = i)}{1 - \mathbb{P}(D \geq S - i)\mathbb{P}(D \geq i + 1)} \times \left(1 - \sum_{k=0}^{i-1} \pi_k(S)\right) + \frac{\mathbb{P}(D = S - i)}{1 - \mathbb{P}(D \geq S - i)\mathbb{P}(D \geq i + 1)} \times \sum_{k=S+1-i}^{S} \pi_k(S)$$

We can then use this distribution, an example of which is shown in Figure 9, to find the service level for various total inventory levels, see Figure 10. Conversely, if we set a service level, we could find a suitable inventory level.

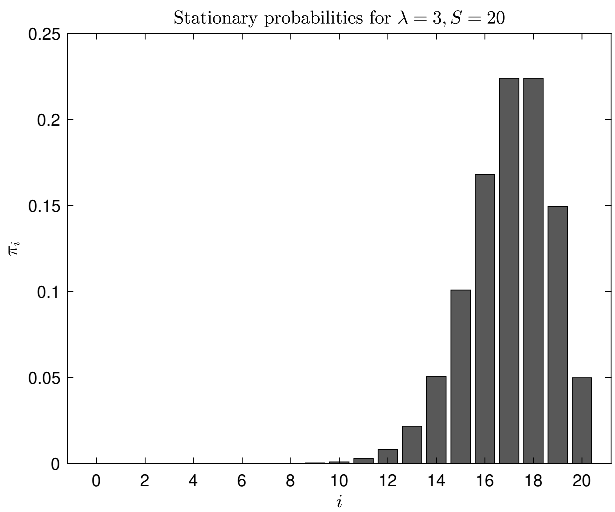


Figure 9: Stationary distribution

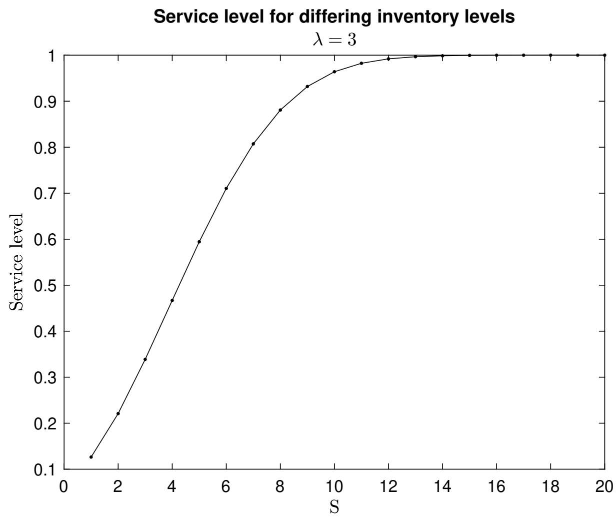


Figure 10: Service level for varying inventory levels

5 Methodology

5.1 Performance indicators

We evaluate the performance of each of the above heuristics using the following Key Performance Indicators (KPIs):
Surgery reschedule rate: the percentage of surgeries that has to be rescheduled because no possible combination of RMD trays that can be used for that surgery is or becomes available within 120 minutes of the scheduled start time, or 30 minutes in case of emergency surgery.
Alternative set rate: the percentage of non-emergency surgeries for which a different combination of RMD trays is used than the one preferred.
Total inventory level: the sum of inventory levels of all tray types.

5.2 Simulations

All simulations were run on the High-Performance Computing (HPC) cluster of the Electrical Engineering, Mathematics and Computer Science (EEMCS) faculty of the University of Twente. We ran an ensemble of 30 identical simulations, parallelised over as many CPU cores, in order to get decent confidence intervals for our KPIs. Each simulation was set up to simulate 728 days of hospital activity, not including 42 days of 'warmup', meant to remove any transient behaviour from the results. The resulting data was then processed in order to obtain the results that we discuss in Section 6.

5.3 Overview of experiments

We performed various numerical experiments using the inventory levels prescribed by the following heuristic configurations:

- Base scenario (current inventory levels)
- Fineman and Kapadia for mean, median and busiest weekday, as described in Section 4.2.
- Base-stock for percentiles ranging from 50 to 95.
- Diamant et al. with 24 hour periods for service levels from 10 to 99 per cent.
- Diamant et al. with 12 hour periods for service levels from 92 to 99.99 per cent.
- Diamant et al. with 8 hour periods for service levels from 92 to 99.9 per cent.
- Diamant et al. with 6, 4 and 2 hour periods for service levels from 92 to 99.999 per cent.

6 Results

An overview of selected simulation results can be found in Figure 11. Right away, we can see that the performance of the different heuristics varies quite significantly. The difference in performance between variants of the same heuristic also seems significant. We discuss the performance of the individual heuristics in Sections 6.1, 6.2 and 6.3. Finally, we comment on the differences between heuristics in Section 6.4).

6.1 Base-stock heuristic

Table 1: Results for base-stock heuristic

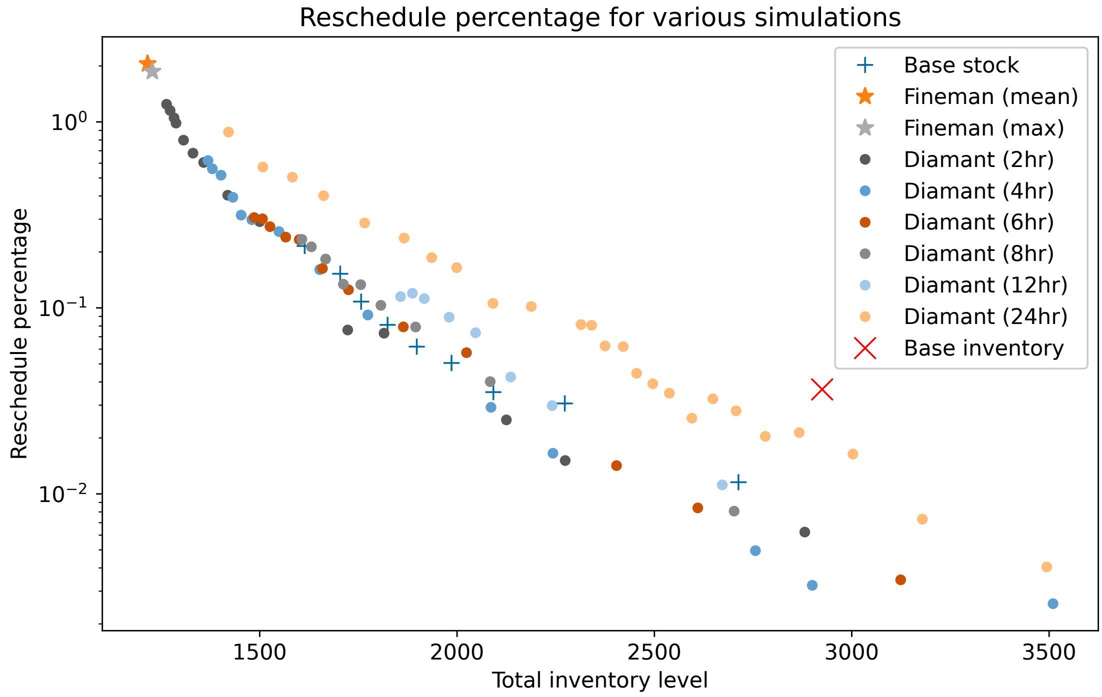
	total inv.	reschedule rate	alt. set rate
Base	2925	0.0376 ± 0.0006	1.196 ± 0.007
BS50	1614	0.214 ± 0.002	6.43 ± 0.02
BS60	1704	0.150 ± 0.002	4.72 ± 0.03
BS65	1757	0.114 ± 0.002	3.979 ± 0.015
BS70	1824	0.081 ± 0.002	3.03 ± 0.02
BS75	1898	0.0605 ± 0.0009	2.44 ± 0.04
BS80	1986	0.0511 ± 0.0010	1.942 ± 0.009
BS85	2092	0.0337 ± 0.0006	1.379 ± 0.014
BS90	2273	0.0329 ± 0.0007	0.869 ± 0.009
BS95	2713	0.0098 ± 0.0005	0.384 ± 0.007

As can be seen in Table 1 and previously shown by Hosteins et al. [3], the base-stock heuristic performs significantly better than the current inventory policy at the Diakonessenhuis. Take, for example, the BS85 instance - it has a lower reschedule rate than the current level, while operating on approximately 28% less inventory.
As Figure 12a shows, inventory levels can safely be lowered for many tray types while maintaining similar overall performance by increasing the inventory in a relatively low (approx. 10%) number of tray types, mostly in less frequently used inventory (Cluster 1) and inventory used in outpatient clinics (Cluster 2).

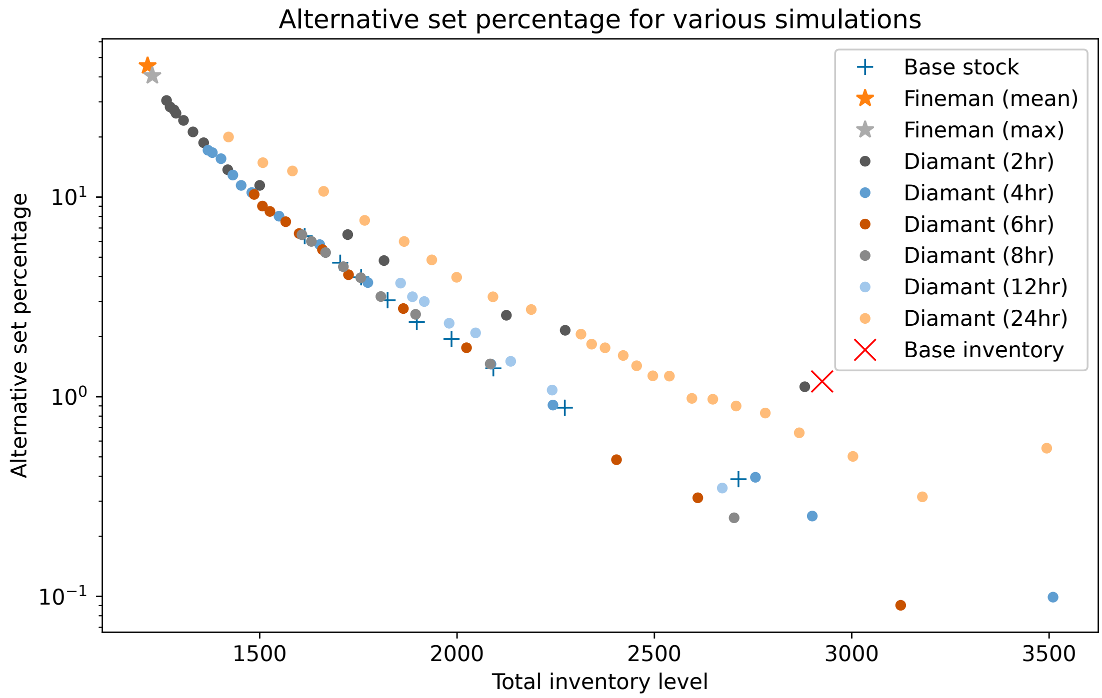
6.2 Fineman and Kapadia

Table 2: Results for Fineman-Kapadia heuristic

	total inv.	reschedule rate	alt. set rate
Base	2925	0.0376 ± 0.0006	1.196 ± 0.007
FKmed	1217	2.070 ± 0.009	45.64 ± 0.03
FKmean	1216	2.054 ± 0.003	45.44 ± 0.03
FKmax	1228	1.859 ± 0.013	40.495 ± 0.018

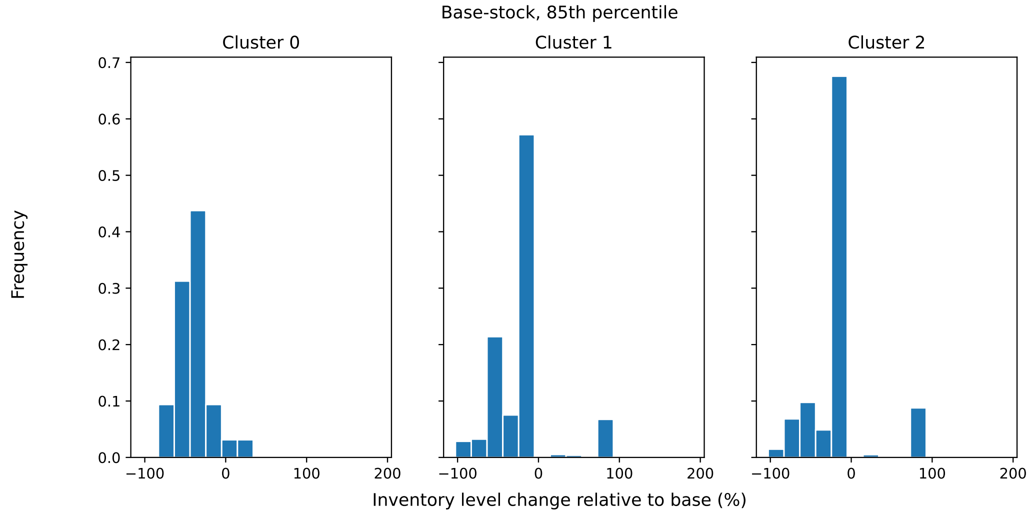


(a) Reschedules

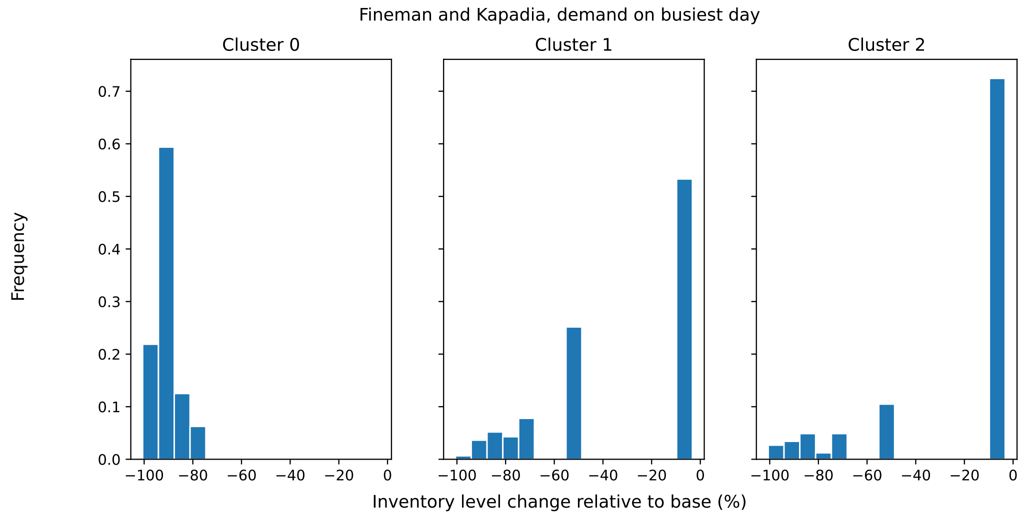


(b) Alternative sets

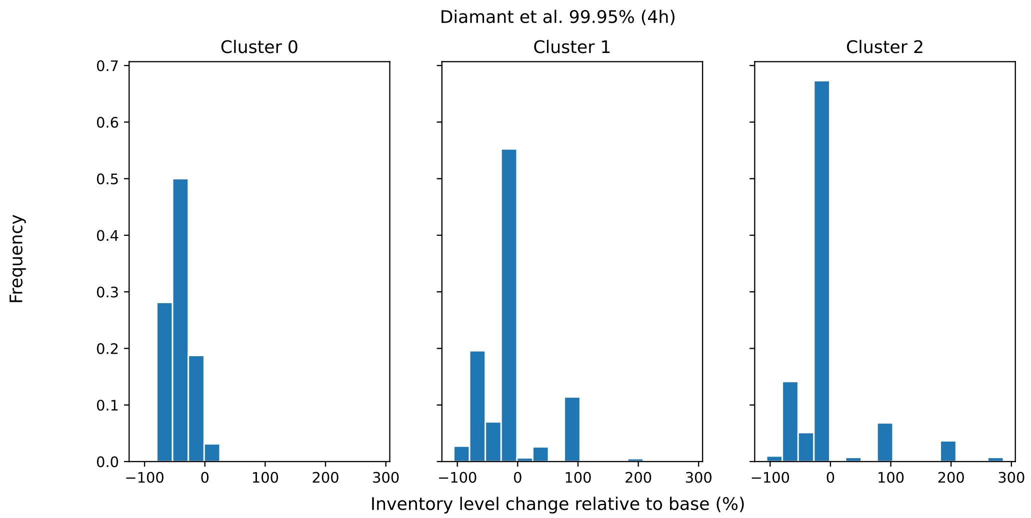
Figure 11: Overview of selected simulation results



(a) Base-stock heuristic (85th percentile) change per cluster



(b) Fineman and Kapadia (max) change per cluster



(c) Diamant et al. 99.95% (4h) change per cluster

Figure 12: Relative inventory change per cluster

Three instances of the Fineman-Kapadia heuristic were tested:

- **FKmed** based on median of the demand for each day of the week.
- **FKmean** based on the average demand over the entire week.
- **FKmax** based on demand on the busiest day of the week.

As Table 2 shows, there is little difference between the **FKmed** and **FKmean** instances, as one might have expected. Interestingly, the **FKmax** heuristic performs significantly better with only just over ten trays more than **FKmed** and **FKmean**. Looking at Figure 11, we can see that this improvement is in line with the improvement shown by other heuristics at similar inventory levels.

6.3 Diamant et al.

Table 3: Results for Diamant et al. heuristic (4 hour period)

	total inv.	reschedule rate	alt. set rate
Base	2925	0.0376 ± 0.0006	1.196 ± 0.007
D92	1369	0.635 ± 0.005	17.17 ± 0.04
D93	1380	0.565 ± 0.005	16.703 ± 0.018
D94	1402	0.511 ± 0.003	15.53 ± 0.02
D95	1432	0.397 ± 0.005	12.84 ± 0.04
D96	1453	0.315 ± 0.002	11.47 ± 0.05
D97	1480	0.2999 ± 0.0012	10.47 ± 0.03
D98	1549	0.251 ± 0.002	7.968 ± 0.019
D99	1652	0.1623 ± 0.0006	5.79 ± 0.02
D99.5	1774	0.0913 ± 0.0011	3.64 ± 0.04
D99.9	2086	0.0280 ± 0.0009	1.456 ± 0.016
D99.95	2243	0.0152 ± 0.0004	0.875 ± 0.014
D99.99	2756	0.0040 ± 0.0004	0.376 ± 0.010
D99.995	2900	0.0031 ± 0.0002	0.268 ± 0.008
D99.999	3510	0.0022 ± 0.0002	0.098 ± 0.005

6.3.1 Period comparison

As we touched upon before, the choice of period is quite a critical one when using the Diamant et al. heuristic. The results in Figure 11 demonstrate this nicely: it is especially obvious how much worse the instances with a 24h period perform when compared to other periods. We see that the 2, 4 and 6 hour period instances perform similarly when it comes to reschedule percentage, with the 8 and 12 hour periods performing slightly worse. When it comes to alternative set rates, the two hour period at points performs worse than even the 24 hour period for similar inventory levels, while the other periods all seem to perform similarly. Given that the median tray sterilisation time is 3 hours and 18 minutes, it is no surprise that methods with periods that match that closely generally perform well, since they most closely conform to reality.

6.3.2 Overall performance

Table 3 shows the results for several instances of the Diamant et al. heuristic with four-hour period, with varying service levels ranging from 92% up to 99.999%. We can see that this performs very well. Take, for instance, **D99.9**, which has slightly less reschedules, but more importantly 40% less inventory than the base scenario. Or take **D99.999**, which has slightly less inventory, but less than one-tenth the reschedule and alternative set rates of the base scenario.

6.4 Comparison

6.4.1 Performance

When comparing heuristics, we look back to Section 1.3, where we specified that we were looking for either
1) Reduction of required stock (at a similar level of performance), or
2) Reduction of unavailability (at a similar inventory level).
To compare with the first criterion, we look at Table 4, where we see that the Diamant heuristic with four hour period and base-stock heuristic are very close - compare, for example, **BS85** and **D99.9 4h**, which have similar total inventory and performance levels.
To compare for the second criterion, we look at Figure 11, where we see the Diamant heuristic with four hour period outperform all others in reschedule rate, while the 6 and 8 hour period Diamant heuristic seem to perform better in alternative set rate. The base-stock heuristic also seems like it would perform quite well here, although our testing did not take it quite up to the inventory level needed to state this conclusively. Finally, we note that we cannot properly compare the Fineman and Kapadia heuristic with the other heuristics - performance seems in line with very low service levels for the Diamant heuristic, so it does not meaningfully perform worse in that aspect - it simply lacks the level of control that we have in other heuristics, as we discuss in Section 6.4.2.

6.4.2 What makes a good heuristic?

When evaluating heuristics, we have to consider more factors than how well the output performs. After all, a heuristic with all-right performance and short run times will more often than not be preferred over one with excellent performance at the cost of a significant amount of run-time. While run time is not

Table 4: Comparison: similar performance level

		total inv.	reschedule rate	alt. set rate
Base		2925	0.0376 ± 0.0006	1.196 ± 0.007
BS85		2092	0.0337 ± 0.0006	1.379 ± 0.014
BS90		2273	0.0329 ± 0.0007	0.869 ± 0.009
D99.99	**2h**	2125	0.0241 ± 0.0004	2.43 ± 0.04
D99.995	**2h**	2274	0.0150 ± 0.0004	2.24 ± 0.04
D99.999	**2h**	2881	0.0056 ± 0.0004	1.13 ± 0.04
D99.9	**4h**	2086	0.0280 ± 0.0009	1.456 ± 0.016
D99.95	**4h**	2243	0.0152 ± 0.0004	0.875 ± 0.014
D99.5	**6h**	2024	0.054 ± 0.002	1.78 ± 0.03
D99.9	**6h**	2404	0.0124 ± 0.0012	0.513 ± 0.011
D99	**8h**	2084	0.0414 ± 0.0007	1.14 ± 0.01
D98	**12h**	2241	0.0288 ± 0.0007	1.100 ± 0.014
D99	**12h**	2442	0.0203 ± 0.0007	0.644 ± 0.003
D91	**24h**	2538	0.0351 ± 0.0005	1.264 ± 0.007
D92	**24h**	2595	0.0246 ± 0.0007	0.998 ± 0.008

problematic for any of the heuristics that we discuss here, there are certain factors that we must weigh, namely **control**, **complexity** and **insight**, which we discuss one by one:

Control: the proverbial number of dials that we can turn in order to influence the final solution, allowing the end user to make decisions, for example in weighing off reschedule rate to total inventory levels. Clearly, this is not the strong suit of the Fineman-Kapadia heuristic, which does not allow its user any real freedom to choose. The base-stock and Diamant et al. heuristics both allow the user to tweak some percentage - service level for Diamant et al, demand percentile for the base stock heuristic. Both these heuristics are therefore to be preferred over the Fineman-Kapadia heuristic when it comes to control.

Complexity: how hard is it for managerial staff to understand and implement? Clearly, the Fineman-Kapadia heuristic is the easiest to comprehend - taking two basic properties and multiplying them is simple. The base-stock heuristic is also easy to understand for anyone who has had an introductory course in statistics. The Diamant et al. heuristic is harder to fully comprehend, using slightly more complex concepts like discrete-time Markov chains. The resulting formulas, however, could be understood by someone who has had an introductory course in statistics, similar to the base-stock heuristic.

Insight: what does this heuristic tell us about the nature of our system? Sometimes, heuristics appear to be arbitrary rules, that do not provide any insight. Luckily, this is not the case for the heuristics we have selected - each one tells us something about the nature of the system. The Fineman-Kapadia heuristic tells us that we must look at the demand within one processing cycle, the base-stock heuristic tells us that the distribution of demand matters, and the Diamant et al. heuristic combines these concepts to paint a more complete picture. With insight, of course, often comes complexity, but also performance. It is our task to weigh these factors against each other, which we do in our concluding remarks (see Section 8).

7 Extensions

7.1 Equalising demand

In Section 2.3.1, we discussed the variability of the demand, and noted that it poses somewhat of a challenge for us. As an experiment, let us see what might happen if we had full control over the distribution of demand over the days of the week. Clearly, it would be best for us if we could spread out demand as much as possible. To illustrate the potential impact of scheduling changes, we ran a simulation with the trays from the D99.95 (4h) instance and with the demand for every weekday set to the mean. The results can be found in Table 5. When demand is equalised, the reschedule rate drops threefold, while the alternative set rate decreases by an entire order of magnitude, which tells us that there are potential gains to be had in performance or inventory levels when inventory levels are weighed in when scheduling. Of course, this scenario represents an unfeasible extreme - we treat it merely for illustrative purposes.

7.2 Tray grouping

As mentioned in Section 2.1, it can happen that a different set of trays is used for a specific procedure, which made us wonder if accounting for this

Table 5: Results for the extensions

type	total inv.	reschedule rate	alt. set rate
D99.9 4h normal	2086	0.0280 ± 0.0009	1.456 ± 0.016
D99.95 4h normal	2243	0.0152 ± 0.0004	0.875 ± 0.014
D99.95 4h equalised demand	2243	0.0051 ± 0.0004	0.0834 ± 0.0008
D99.95 4h grouping	2208	0.08488 ± 0.00013	7.412 ± 0.018

effect might allow for lower total inventory levels at the cost of higher alternative set rates by computing inventory levels for groups of 'compatible' tray types together. One might question, of course, whether this is a trade one would want to make - we endeavoured merely to find if it is in any way effective. This required us to first develop some measure of 'compatibility' between tray types, based on limited data and without the involvement of subject matter experts (SMEs). We had at our disposal a list of all configurations of tray types that were used for each kind of surgery during 28 weeks in 2021, from the data acquired by Hosteins et al. [3]. Outpatient procedures use only one type of tray each, so these were excluded. This list is structured as follows (excerpt from data for one of the surgery types):

[[153], [148, 205], [205, 148], [148, 205], [148, 205], [148, 205], [148, 155, 153, 205], [148], ...],

where each number represents a tray type. As we can see, this list has not been deduplicated, allowing us to find the relative frequency of each of the configurations listed.
To obtain some measure of compatibility from this data, we performed a series of steps that we detail in Appendix A. Once we found a measure of compatibility between tray types, we set a threshold for compatibility, and created a graph G where the vertices represent tray types and edges are created between compatible (that is, compatibility above a certain threshold level, see Appendix A) tray types. In this graph, we then find all the cliques. Starting at the largest clique, we select cliques in such a way that no two selected cliques have a vertex in common. We then sum of the demand for all tray types in a clique, and calculate the total inventory I for each clique using the Diamant et al. heuristic with a service level of 99.95% and a period of 4 hours. We then distribute the total inventory over the different tray types as follows: first, we calculate for tray type t what fraction f_t of the total demand for that clique originates with this type. We then set the inventory level of the tray type I_t to $I_t = \lfloor f_t \cdot I \rfloor$. If $\sum I_t < I$, we go down the list of tray types in decreasing order of demand, increasing the inventory level by one, until $I = \sum I_t$.
The results of this grouping can be found in Table 5. As expected, we see the alternative set rate increase significantly. We also see the reschedule rate rise significantly. A comparison with the results of the Diamant et al. 99.9% (4h) results shows that a reduction of inventory level can be attained with much lower loss of performance, which means that we cannot recommend this specific version of tray grouping. We cannot rule out, however, that a similar scheme may perform much better than we show here, given our limited experimentation.

8 Concluding Remarks

8.1 Summary

In this report, we looked into the problem of finding a heuristic for inventory levels of RMD trays. We found that various techniques from traditional inventory theory do not apply. We tried three heuristics from domain-specific literature, and found that the heuristic proposed by Diamant et al. [2] with a four-hour period performs best. We found that the base-stock heuristic proposed by Hosteins et al. [3] performs remarkably well for its simplicity. The heuristic based on the work of Fineman and Kapadia [1] performs quite well relative to inventory level, but lacks any semblance of control, thereby making it unsuitable for our purpose. We also looked at two extensions: demand equalisation and tray grouping. Demand equalisation showed great potential for improving performance at the same inventory level, or reducing inventory levels at the same performance level. We were not able to make tray grouping work well.

8.2 Recommendations

For best results, the Diakonessenhuis should consider using the heuristic from Diamant et al. with a proper period, matched to the time it takes to sterilise the RMDs to set new inventory levels. The base-stock heuristic could also be used, if they prefer trading some performance for simplicity. They might also consider taking RMD inventory into account when planning surgeries, though further research is needed in this area. They might also consider looking critically at the number of different tray types, to see if some types could not be consolidated in order to make use of the pooling advantage.

Some variation of the algorithm described in Section 7.2 could still be of use here to indicate where consolidation might be possible. If not, they should at least investigate the tray types with zero demand, as mentioned in Section 2.3.1.

8.3 Future research

There are several avenues of future research that seem promising:
Firstly, research could focus on developing better heuristics for the RMD tray inventory level problem. One might, for example, create some kind of iterative method based on discrete-event simulation that updates the inventory levels after every run until they converge to optimal values (though this would be a computationally costly method). Secondly, research could be done into weighing material constraints like RMD tray inventory in the already complicated process of operating theatre planning. Finally, more research could be done on optimising the composition of RMD trays (see for example van de Klundert et al. [15]), which has the potential to drastically reduce the number of required RMDs [16]. Of course

9 Reflection

While reflecting on the process of writing this bachelor thesis, the author was reminded of the old adagium “Hindsight is 20/20”. Especially in the exploratory phase, much time was spent on ideas and methods that, as it turned out later, were not worth the effort. While some of this is a natural part of the process, it is the author’s conviction that a significant part of this could have been prevented, had he been more thorough in his review of potential heuristics. Looking back, one might also conclude that communication was not always timely, though this luckily did not result in any major delays. These points are, however, overshadowed by a certain sense of accomplishment, now that this report is complete. Instead of looking back, the author now wishes to look forward, towards the coming year, when, God and committee willing, he will be starting his master’s in Applied Mathematics.

References

[1] S. Fineman and A. Kapadia, “An analysis of the logistics of supplying and processing sterilized items in hospitals,” *Computers and Operations Research*, vol. 5, no. 1, pp. 47–54, 1978.

[2] A. Diamant, J. Milner, F. Quereshy, and B. Xu, “Inventory management of reusable surgical supplies,” *Health Care Management Science*, vol. 21, pp. 439–459, Sept. 2018. Publisher: Springer Nature.

[3] G. Hosteins, H. Bos, and G. Leeftink, “Improving Hospital Sterilisation Processes: a Comprehensive Simulation Model of the Integrated Reusable Medical Devices Cycle.” CHOIR, University of Twente, [manuscript submitted for publication], 2023.

[4] OECD, “Health Status,” 2021. Available at `https://stats.oecd.org/Index.aspx?DatasetCode=HEALTH_STAT`, last visited 2023-04-05.

[5] L. William Traverso and K. Hargrave, “A prospective cost analysis of laparoscopic cholecystectomy,” *The American Journal of Surgery*, vol. 169, pp. 503–506, May 1995.

[6] Frost & Sullivan, “Driving operational improvement in hospitals through an advanced instrument management system.” Available at `https://www.bd.com/documents/case-studies/surgical-instruments/SU_Frost-Sullivan_CS_EN.pdf`, last visited 2023-05-04.

[7] A. M. de Bruin, R. Bekker, L. van Zanten, and G. M. Koole, “Dimensioning hospital wards using the Erlang loss model,” *Annals of Operations Research*, vol. 178, pp. 23–43, July 2010.

[8] E. Akçalı and S. Çetinkaya, “Quantitative models for inventory and production planning in closed-loop supply chains,” *International Journal of Production Research*, vol. 49, pp. 2373–2407, Apr. 2011.

[9] R. Dekker, M. Fleischmann, K. Inderfurth, and L. N. Van Wassenhove, eds., *Reverse Logistics*. Berlin, Heidelberg: Springer, 2004.

[10] H. M. Wagner and T. M. Whitin, “Dynamic Version of the Economic Lot Size Model,” *Management Science*, vol. 5, pp. 89–96, Oct. 1958.

[11] K. Richter and M. Sombrutzki, “Remanufacturing planning for the reverse Wagner/Whitin models,” *European Journal of Operational Research*, vol. 121, pp. 304–315, Mar. 2000.

[12] M. Bijvank and I. F. A. Vis, “Inventory control for point-of-use locations in hospitals,” *Journal of the Operational Research Society*, vol. 63, pp. 497–510, Apr. 2012.

[13] M. Di Mascolo and A. Gouin, “A generic simulation model to assess the performance of sterilization services in health establishments,” *Health Care Management Science*, vol. 16, pp. 45–61, Mar. 2013.

[14] B. Rupnik, R. Narding, and T. Kramberger, “Discrete event simulation of hospital sterilization logistics,” *Tehnicki Vjesnik*, vol. 26, no. 5, pp. 1486–1491, 2019.

[15] J. Van De Klundert, P. Muls, and M. Schadd, "Optimizing sterilization logistics in hospitals," *Health Care Management Science*, vol. 11, no. 1, pp. 23–33, 2008.

[16] M. Vermunt, "Instrumentennetten op maat; duurzaam en tóch veilig? [Custom RMD trays; durable, but still safe?]," May 2021. Available at `https://atosborne.nl/nieuws/instrumentennetten-op-maat-duurzaam-en-toch-veilig/`, last visited 2023-03-30.

A Tray grouping procedure

Here, we explain our procedure for grouping tray types. We start with data from surgeries, where for each surgery we have a list containing all tray type configurations used during a certain time period. We work out an example based on the following list, taken from the data set:

[[153], [148, 205], [205, 148], [148, 205], [148, 205], [148, 205], [148, 155, 153, 205], [148], ...],

where each number represents a tray type. To get from this list to some measure of compatibility, the first step we take is to transform this list into a tree structure, using Algorithm 1. First, we find the number of occurences for each tray type. The results can be found in Table 6.

Table 6: Type counts for configuration list

Tray type	**Count**
148	146
205	138
153	37
155	4
156	1
21	1

We then sort the trays in the configurations in descending order of occurrence:

[[153], [148, 205], [148, 205], [148, 205], [148, 205], [148, 205], [148, 205, 153, 155], [148], ...]

We then add each of the configurations to a tree, as described in Algorithm 1. The result can be found in Figure 13.

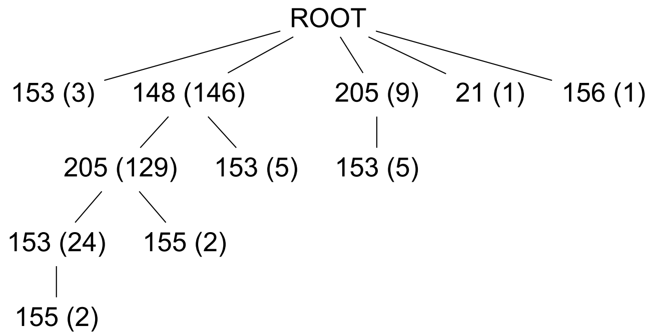


Figure 13: Tree generated from configuration list. Key: Tray type (occurrences)

This tree is a compact representation of all the configurations used in a given surgery. One can read it as follows: starting from the root node, every step along the tree adds a tray type to the configuration. The number in parentheses indicates the number of occurrences of the current sequence, including configurations with more tray types. For example, going from the root node down to 148, then down to 205 and finally to 153, we get the configuration [148, 205, 153]. Configurations including this sequence occur 24 times - this includes the two occurrences of [148, 205, 153, 155].

Besides giving insight into the frequency of various configurations, the tree gives insight into which tray types might be interchangeable - these are the types which occur as sibling nodes (i.e., nodes that have the same parent node) in this tree. Of course, if we want to ensure that a tray type can be exchanged for another quite frequently, we need to count both when it occurs with other trays as sibling nodes and when it does not - not only for one surgery (like in this tree), but for all of them.

We now give an example of the application of Algorithm 2. We first descend down the tree to (ROOT, 148, 205, 153). We note that this node has only one child, and that it occurs here twice. Now that we have counted these occurrences, we can remove them from this child and its parent nodes (153, 205 and 148). This leaves us with the tree that can be found in Figure 14.

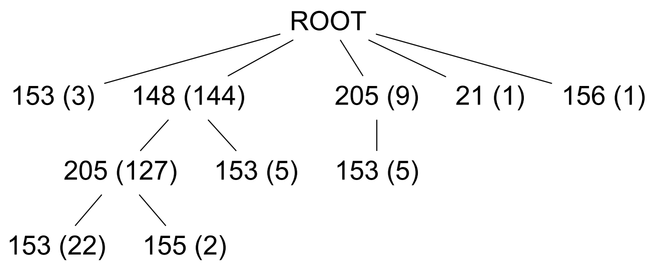


Figure 14: Tree after first count.
For simplicity, nodes with zero count are left out. Key: Tray type (occurrences)

We now move up a level to (ROOT, 148, 205). We note that this has two children: 153, occurring 22

Algorithm 1 Tray configuration list to tree structure

Input:
list of tray configurations for a surgery $\mathcal{C}$.
Output:
A tree $\mathcal{T}$, where each node $n \in \mathcal{T}$ stores a tray type number and a counter.
A list $\mathcal{L}_{\text{occur}}$ containing the number of occurrences of each tray type.

1: **Initialise** an empty tree $\mathcal{T}$ with a root node n_{root}
2: Store the number of occurrences of each tray type in a list $\mathcal{L}_{\text{occur}}$
3: **for** $c \in \mathcal{C}$ **do:**
3.1: Sort c by $\mathcal{L}$ in descending order
3.2: Set the current node n_{current} to n_{root}.
3.3: **for** tray type $t \in c$ **do**:
3.3.1: If not present, add a child node n_{child} of n with tray number t and counter at 1 to $\mathcal{T}$.
3.3.2: Else, increment the counter of the child node n_{child} by 1.
3.3.3: Set n_{current} ot n_{child}.

Algorithm 2 Counting co-occurences

Input:
Tree structure $\mathcal{T}$, where each node $n \in \mathcal{T}$ stores a tray type number and a counter.
Number of total tray types N.
Output:
A sparse matrix M_{config} of size $N \times N$,
containing the number of tray configurations in which trays co-occur.
A sparse matrix M_{occur} of size $N \times N$,
containing the number of co-occurences of each combination of trays.

1: **Initialise** M_{config} and M_{occur} as $N \times N$ sparse matrices.
2: **Apply Subprocedure 1** to the root node n_{root} of $\mathcal{T}$.
3: **Subprocedure 1**
Input: current node $n \in \mathcal{T}$, root node n_{root} of $\mathcal{T}$.
3.1: **if** n has children **do:**
3.2: **for** each child n_{child} of n **do:**
3.2.1 **Apply Subprocedure 1** to n_{child}.
3.3: **for** each pair of children (n_1, n_2) in the children of n **do:**
3.3.1: Add 1 to $M_{\text{config}}[t_1, t_2]$, where t_1 and t_2 are the tray numbers stored in n_1 and n_2, respectively.
3.3.2: Add the count of n_1 to $M_{\text{occur}}[t_1, t_2]$, where t_1 and t_2 are like in 3.2.1.
3.4 **else do:**
3.5 Recursively subtract the count stored in n from the count stored in each of its parent nodes, excluding n_{root}.

times, and 155, which occurs twice. We note that we saw 153 and 155 occur together, subtract the occurrences as before, and move up a level to (ROOT, 148).

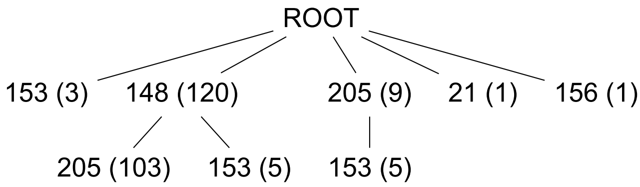


Figure 15: Tree after second count.
For simplicity, nodes with zero count are left out.
Key: Tray type (occurrences)

Again, we note the occurrences of 205 and 153 at the same level, and subtract the number of occurrences of 205 and 153 from those of 148. We move over from (ROOT, 148) to (ROOT, 205) and note that it has an 'only child' 153, and subtract its count from the number of occurences of (ROOT, 205). Finally, we note all of the children of the root node and their co-occurrences. Finally, of course, we move up to the root node, repeating the same procedure for each of its children as we have for the children of 148. In the end, we find matrices M_{config} containing the number of tray configurations in which tray types co-occur (Table 7) and M_{occur} containing the number of co-occurences of each combination of trays (Table 8).

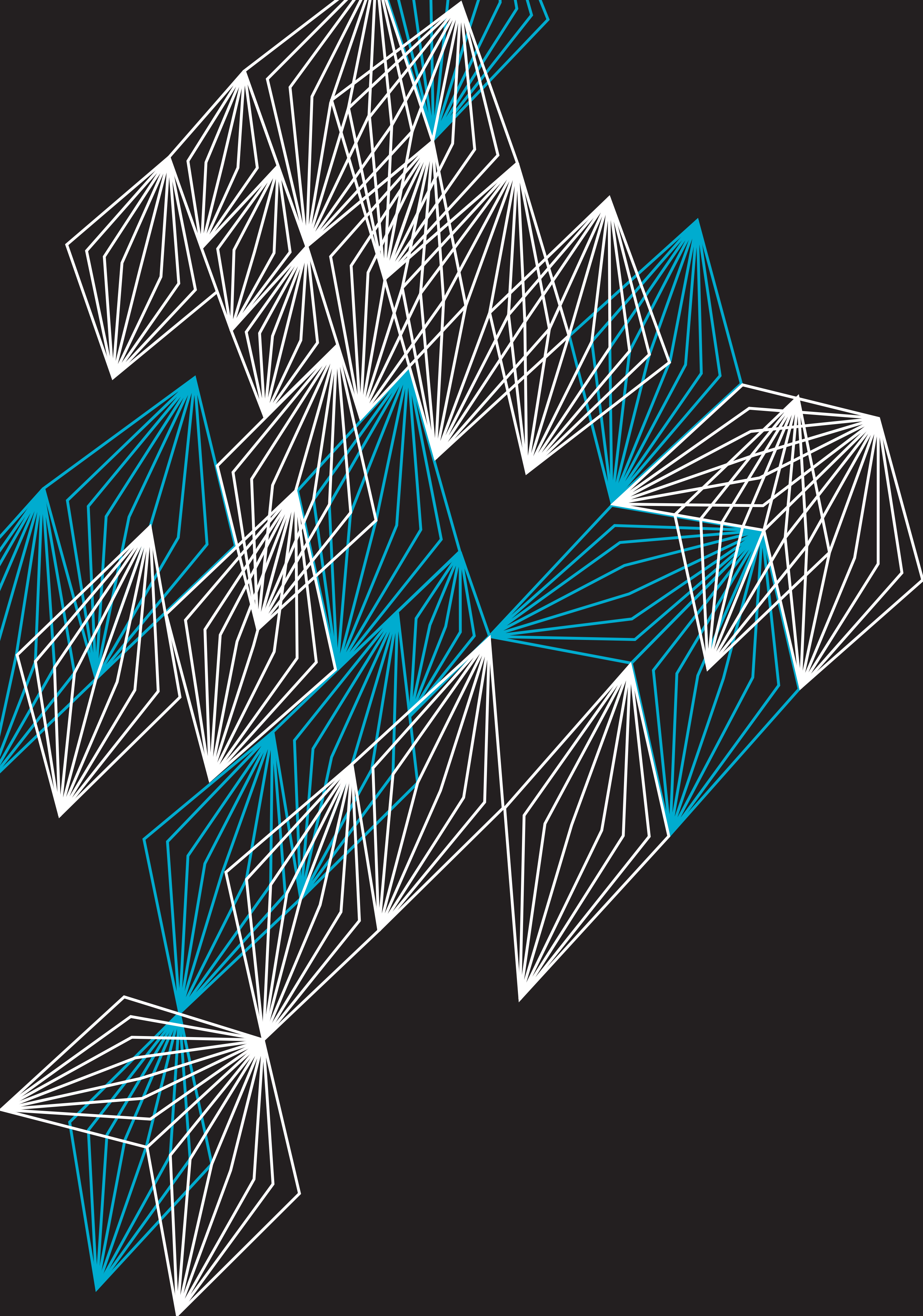
Table 7: M_{config} (all-zero rows/columns left out)

	148	205	153	155	156	21
148	1	1	1	0	1	1
205	1	2	2	0	1	1
153	1	2	4	1	1	1
155	0	0	1	2	0	0
156	1	1	1	0	1	1
21	1	1	1	0	1	1

Table 8: M_{occur} (all-zero rows/columns left out)

	148	205	153	155	156	21
148	12	12	12	0	12	12
205	4	107	107	0	4	4
153	3	22	35	1	3	3
155	0	0	2	4	0	0
156	1	1	1	0	1	1
21	1	1	1	0	1	1

We can now repeat this process for each type of surgery, each time adding to $M_{\text{config}}, M_{\text{occur}}$ and $\mathcal{L}_{\text{occur}}$. Using this data, we can define a number of different measures of compatibility. We tried various measures, and settled on the relative number of co-occurences of each combination of trays (that is, every column in M_{occur} divided by the list from Table 6, which we named $\mathcal{L}_{\text{occur}}$ in Algorithm 1). We set a threshold of 0.8 for this metric, and used this to create a graph as described in Section 7.2.



UNIVERSITY OF TWENTE.