ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Bachelor Thesis Econometrics and Operational Research

Markov-modulated analysis of a spare parts system with priority delivery and changes to order and holding costs

Name student: Bert Helderman
Student ID number: 453689

Supervisor:W. Ma
Second assessor: R. Dekker

Date final version: 7 July 2019

Contents

Abstract

Capital goods often require new parts as their life cycle comes to an end. The fact that suppliers for the spare parts required stop producing them and the long distance from production to repair facility together result in random lead times and disruptions. Shortages are very costly with regard to capital products thus it is important for any company that these are minimised. We analyse this as an inventory system and change certain cost parameters. We follow up by calculating the effects of each change and incorporating them into cost functions found in literature. We find cost deviations of up to 60% as the effects of random lead times and disruptions. The changes in cost parameters each result in a small cost deviation of up to 15%.

1 Introduction

Capital goods such as planes, ships or heavy machinery can last a long time before decommission. Because of the size of these goods, it is usually cheaper to replace a broken part than to replace the entire product, even when many other parts are worn out. However, when products near the end of their lifetime, suppliers of these spare parts often stop production or delay deliveries in favour of customers with newer capital goods, as these will remain customers for longer and thus be more profitable. This means that e.g. airlines with older planes face an increasing risk of supply-side disruption when spare parts are necessary, which may cost millions. This risk comes from multiple sources. A supplier for this type of low-demand parts usually has production located in a single factory, far from where the parts are required. The long transportation time results in random lead times. Furthermore, problems may occur during production, making ordering and shipping parts temporarily unavailable. Therefore, it is interesting but difficult to determine optimal levels of inventory for the airline company for this type of good.

The research done by Hekimoglu et al. (2018) uses a Markov-modulated supply system with random lead times to model this problem. They prove a base stock policy is optimal, but these optimal stock levels are obviously dependent on order, holding and shortage costs. This means an analysis of the effect of different costs is also required.

This paper aims to implement their methods and subsequently check their results through simulation. We will start by explaining relevant literature in Section 2. Then we continue with detailing our implementation of the work of Hekimoglu et al. (2018) in Section 3, which is followed by comparing the results. Furthermore, we will extend the work of Hekimoglu et al. (2018) by changing the assumption of constant cost parameters. We report on this in Section 4. Finally, we summarise our findings in Section 7.

2 Literature review

We start with a description of inventory systems as is commonly used in literature. They have a number of basic properties, such as demand, lead times, acquisition, shortage and holding costs, and planning horizon. Each period of the planning horizon the company orders a number of parts given their expectation of future demand and pays the acquisition costs of the order, which will be delivered in a number of periods (lead time). Then the demand of this period is subtracted from the current inventory level, after which shortage or holding costs are incurred.

In general costs per unit per period are taken to be constant throughout the planning period, which is a reasonable assumption for shorter planning horizons. Demand is normally considered to be random, since (almost) fixed demand is only present in very specific situations, such as fuel transported through pipelines. Lead times are taken fixed or random according to the research goals of a paper.

There is much research concerning inventory systems with supply disruptions or random lead times, but only one paper concerning both: Hekimoglu et al. (2018). They perform an analysis of a spare part inventory system with supply disruptions, random lead times and no crossovers between orders. With these assumptions, they create an optimal ordering policy given the disruption state of the supplier. Furthermore, they investigate a number of cases regarding the progression of supplier health (increasing or decreasing over time) and whether disruptions were short and frequent (SFD) or long and infrequent (LID). They show that supply disruptions and random lead times have a combined effect that is extremely significant when supplier health decreases over time.

In this paper we also deal with changes in order cost and priority delivery. Earlier research by Alfredsson and Verrijdt (1999) has proven that an option for priority delivery when stock-outs occur significantly reduces costs, although they use shipments between warehouses or from a central production plant. There appears to be little research regarding occasional order cost changes in a finite horizon inventory system, as the closest we can find was done by Lev and Weiss (1990). They develop an optimal policy for the economic order quantity (EOQ) model, given any order cost changes. However, the model studied in this paper differs significantly from the EOQ model, so their results cannot be used here. Therefore we appear the first to study Markov-dependent order cost changes in an inventory system.

3 Basic problem

In this section, we replicate the methods of Hekimoglu et al. (2018) in order to validate their results and establish a benchmark to compare our extensions to. We start with re-iterating the methodology of Hekimoglu et al. (2018), motivating how we filled a few small gaps in their methods. We may refer to Hekimoglu et al. (2018) as 'the paper' in the next sections.

3.1 Problem description

The paper details a single-item inventory system, with relatively low demand (D) per period. The supplier of this inventory system has a number of distinct states $i \in B$, which can be divided into healthy states and disruption states. They assume an order to the states, with higher state number indicating a worse state (longer lead times and higher disruption chance). When the supplier is healthy, orders can be placed and delivered as normal, but when the supplier is in disruption, no new orders can be placed. Each healthy state has a separate disruption state d^i, with their own disruption ($q(i)$)and recovery probabilities ($\zeta(i)$). This system creates supply-side risk, but does not create random lead times. To model this, the paper gives each supply state a different probability of delivering orders ($b(i)$) in any given period.

They use two semi-dependent queues to model Markov-modulated random lead times. The first queue has a number of items in it equal to the state the system finds itself in. If the supply stays healthy in a period, there is a probability e to have an item arrive and probability d to have an item leave this queue. When $e > d$, we have an unstable supply, while $e < d$ indicates stable supply. The number of items is equal to the state number, with higher number of items indicating worse supply health. Then we have the second queue, which contains all previously placed orders and has a partial-batch server with batch size equal to the capacity of the queue. When the queue is full, new orders are added to the last item in the queue. The service rate of this queue, $b(i)$, is dependent on the number of items i in the first queue. Since the server has capacity equal to the size of the queue, it is sufficient to set the capacity to two. Together this creates a Markov-modulated random lead time without order crossovers. Below we show these queues for $i = 6$, 7 outstanding orders and a capacity of 10.

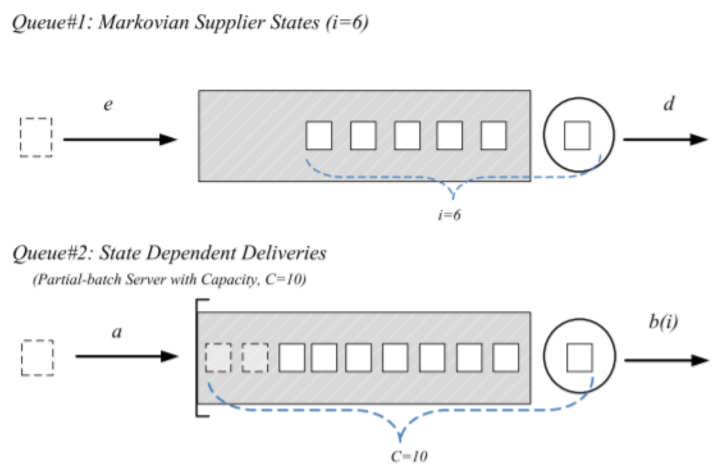

Figure 1: Example of queue 1 and 2. From Hekimoglu et al. (2018), section 4.5

Please note that in reality, queue 1 is used to create the Markov transition matrix and queue 2 is required only to calculate some probabilities, thus we do not need to model these queues. They are used only as a way to derive the transition matrix and delivery probabilities.

Their method is divided into 2 parts: in the first they give an analytic analysis of the model, where they determine a state-dependent base stock policy is optimal. They also use the value-iteration algorithm to calculate these base stock levels. We use the property of the cost function they found to simplify the process of finding optimal base stock levels. In the second part, they run a number of simulations with different scenarios. These have no supply risk and deterministic lead time, either of them, or both. They run these for 24 different parameter sets, including SFD and LID, unstable and stable supply, different delivery parameters and for 5,10 and 15% of periods in disruption. These parameters and our results can be found in the Appendix. The different scenarios are then used to determine the size of the effects of random lead time, nonstationarity, supply risk, and coupled effects. We expect to find similar results, with slight differences due to the use of simulation. Finally, the paper applies their method on a business case, showing it lowers costs and smooths inventory levels. We explain these methods in further detail below.

3.1.1 Base stock levels

There are some additional parameters to be explained. We start with c, h and p, which are order cost per unit, and holding and shortage costs per unit per period. The letters x and y indicate inventory level and inventory position (inventory level + outstanding orders), respectively. The random variable D_{l+1} stands for the convolution of demand of $l+1$ periods. The random lead time of state i is indicated using $L(i)$, and finally α is the discount rate per period. Base stock levels are calculated with this notation using the following four functions.

$$C^l(x) = \alpha^l \mathbf{E}((h \max(x - D_{l+1}), 0) + p \max(D_{l+1} - x, 0)) \tag{1}$$

$$C(i, x) = \sum_{l \geq 0} Pr\{L(i) \leq l \leq L(i^+)\} C^l(x) \tag{2}$$

$$\begin{aligned} W_n(i, x) = \min\nolimits_{y \geq x}(cy(1 - \alpha q(i)) + (1 - q(i))C(i, y) + q(i)C(d^i, y) \\ + \alpha(1 - q(i))\mathbf{E}(W_{n-1}(i_+, y - D)) + \alpha q(i)\mathbf{E}(g_{n-1}(d^i, y - D))) \end{aligned} \tag{3}$$

$$g_n(d^i, y) = C(d^i, y) + \alpha(1 - \zeta(d^i))\mathbf{E}(g_{n-1}(d^i, y - D)) + \alpha\zeta(i)\mathbf{E}f_{n-1}(d^i_+, y - D) \tag{4}$$

Equation 1 gives the discounted costs for l periods ahead, where either holding or shortage costs are incurred. The next equation sums these costs for all remaining periods in the planning horizon, weighted by the probability that the outstanding orders are delivered in the given period l. Then Equation 3 gives the cost of having base stock level y conditional on the disruption status of the next period, given we are in a healthy state and there are n periods left in the planning horizon. Similarly, Equation 4 gives the cost of having inventory level y while in a disruption state. Equations 3 and 4 are defined recursively. Calculating $Pr(L(i) \leq l \leq L(i_+))$ is done using Algorithm 1 from Hekimoglu et al. (2018), which uses queue 2 with capacity of 2 to find them.

We calculate base stock levels using the value-iteration algorithm for Markov decision processes (MDP). In essence, we determine the optimal value for y each period, working backwards from the end. After we determine this optimal value, which is simple since Hekimoglu et al. (2018) has determined that W is convex in y, we enter this value (y) into the function g to calculate its current value. We set $W_0(i, x)$ and $g_0(d^i, y) = 0$. Then we calculate the value for y which minimises $W_1(i, x)$ using the previous values W_0 and g_0 for the recursive parts of the function. This continues until we reach the first period of the planning horizon.

Each period, the input inventory value for both recursive functions decreases by $\mathbf{E}(D)$. The function $W_{n-1}(i_+, y - D)$ may not have the same value of $W_{n-1}(i_+, y)$, and this applies for function g as well. Fortunately, the input $y - D$ is only a lower bound on the value of y for the next period, thus we do not need to evaluate function W until the end of the planning horizon each period. This is not true for function g, since we cannot place new orders during disruption, which means it needs to be evaluated until the end each period.

Now we explain the simulation used to both determine optimal stock levels and actual costs associated with a certain stock level.

3.1.2 Simulation

This simulation takes a set of stock levels and other parameters as input and then returns a single value of discounted cost. Step 1 uses a different Markov transition matrix if the current scenario does not include disruptions and step 2 calls to different functions depending on whether the current scenario uses random or deterministic lead times. The basic structure thus remains the same no matter which properties are present.

The simulation is divided into n periods, with n the horizon. Each period we have 5 steps:

1. The supplier state changes, possibly to the same state
2. Earlier orders may be delivered
3. If the supplier is healthy, a new order may be placed
4. Demand from this period is subtracted from inventory
5. Discounted costs for this period are incurred

We initialise the simulation in the healthiest state, with inventory equal to the base stock level calculated for this state. We use a horizon of 100 periods (approximately 8 years) and 50000 replications, just like the paper. To control variance, we use common random numbers. Parameters can be found in appendix A.

3.1.3 Base stock levels using simulation

Additionally, we used the above simulation to estimate optimal base stock levels, where optimal means lowest associated costs. In order to estimate optimal stock levels, we first perform an enumeration of the stock levels between 1 and 60 for each healthy state (since orders cannot be placed in disruption states a base stock level is meaningless) and find costs associated with each set of stock levels through the simulation. We use only run the simulation 5 times for each stock level set at first to ensure feasible computation times, as this method requires $O(n^3)$ computation time, where n is the maximum value of stock level. The costs of each set of stock levels is then the mean of these 5 runs. Then we use these stock levels, increased by 5, as new maximum stock levels for each specific combination of parameters, scenario and state. The increase of 5 should be sufficient as the initial enumeration is likely to be close to the optimum and - for most scenario/parameter combinations - stock levels are below 20. Thus an increase of 5 is likely to include the optimal stock level.

Then we run the same simulation again but with the new maximum stock levels and an increased number of iterations of 100. This method ensures we do not run the simulation for a large number of stock levels that are much higher than the optimum. In theory one could repeat this process multiple times to find even better bounds for the stock levels, but we have found this method to be very expensive with only two steps.

We calculate costs for the stock levels found in the simulation and the stock levels calculated using value iteration. We expect no differences, but should we find any, we will default to the simulation stock levels as the probability of logical errors is smaller there.

3.1.4 Scenarios

In order to calculate the effects of random lead time, supply risk and their coupled effect, the paper calculates base stock levels without taking into account one or both of these properties. These are fed into the simulation, which does have these properties. The relative increase in costs, compared to using supply risk and random lead times in the calculation of the stock levels, is considered to be the effect of this property.

There are 4 different scenarios for the 2 properties. The paper uses an additional scenario to calculate the effect of non-stationarity (i.e. the different Markov states) on the costs. This scenario will not take supply risk or random lead time into account, and it will choose the deterministic lead time as equal to the average of the expected lead time of each healthy state.

Using these scenarios, they calculate the relative effects of each property in the following manner:

$$\Delta^c = \Delta^{Nonst} - \Delta^{RLT} - \Delta^D \tag{5}$$

$$\Delta^N = \Delta^{St} - \Delta^{Nonst} \tag{6}$$

Here Δ^c stands for the coupled effect of supply risk and random lead time, Δ^{Nonst} stands for the combined effect of supply risk and random lead times and is calculated as the deviation of the scenario that ignores random lead times and supply risk compared to the scenario that does not ignore these (the base case). The relative increase in costs of ignoring random lead time or supply risk is indicated by Δ^{RLT} and Δ^D, calculated in a similar manner. Then we have the effect of ignoring non-stationarity indicated by Δ^{St} and finally, the effect of non-stationarity in Δ^N.

3.1.5 Business case

The paper runs a simulation with a single healthy and unhealthy state using parameters calculated from historical data. They calculate stock levels associated with 90% and 99% ready rates (percentage of demand immediately fulfilled from stock). We will use the previous simulation to calculate these stock levels to validate their results, but we will not repeat their calculation of other parameters, since we do not have the used data available. We will enumerate stock levels and find costs associated with each value using the same simulation, but without the two-step approach used since we have only 2 states to consider, which changes the method from a $O(n^3)$ to $O(n^2)$ computation time. Therefore we can immediately use the mean costs of 100 simulation iterations to determine optimal stock levels. However, this case also has a service level requirement. Therefore we calculate the mean service level for each set of stock levels at the same time as the costs, and if this does not meet the requirement, this set of stock levels is excluded from being selected as optimal.

Once we find the optimal set of stock levels fulfilling the service level, we run the standard 50000 iterations to find the associated costs.

4 Extended problem

Following the validation of the results of Hekimoglu et al. (2018), we extend their research by changing certain assumptions. We choose to relax the assumption of fixed ordering and holding cost parameters. We motivate and explain each changes in Section 4.1, then follow up with our methods in Section 4.2. This is followed by an overview and discussion of the results in Section 6.

4.1 Motivation and hypotheses

4.1.1 State-dependent order cost

First, we deal with goods that require both large machinery and expertise to produce. This means entering this market is very expensive. Second, for some spare parts there are no real substitutes. These two factors lead to a monopoly spare part service provider in a market, who thus has the power over pricing the spare parts. In the original paper, states with higher numbers indicate worsening supply health, which results in longer lead times and increased chance of disruption. It can also indicate financial problems at the supplier. This can lead to an increase in spare part price to keep the company afloat.

We set a 25% discount on order price in state 1 and a 25% increase in state 3 to keep average price equal to the case where it is ignored. The new vector of order cost is thus [1.5, 2, 2.5]. Our hypothesis is that costs will not be affected much by this change, but we do expect it to smooth stock levels between states, since higher stock levels will be cheaper in state 1 and more expensive in state 3.

4.1.2 Priority delivery

Almost every delivery system has an option for priority delivery, usually at a large increase in cost associated with delivering orders before other customers or using a faster delivery method, such as an air plane instead of a freight ship.

Priority delivery will halve the probability of there being no delivery during a period, which then increases the complementary probability of a delivery occurring in such a way they still add up to 1.

We will implement the decision to use priority delivery in a simple manner: when inventory becomes lower than the expected demand of one period, priority delivery is requested. Then once the next delivery arrives, priority costs are incurred, which are twice that of normal order costs. During normal operations (no disruption) the probability of incurring shortage costs is small as orders can be placed and expected lead time is not high. But when in disruption, no new orders are placed, thus long disruptions can easily lead to shortages. We expect it to be used sparingly right after the end of long disruptions, therefore we predict the effect will be much larger for the LIDs compared to SFDs.

4.1.3 Two-stage holding cost function

Ordinarily, ordered parts are stored in a warehouse until needed. The capacity of such a warehouse is finite. Therefore, the company may have more inventory than the capacity of its warehouses. When this happens, it has to rely on hired storage space, which is more expensive. While one could say the company should build enough storage space to hold its base stock levels worth of parts, this is not necessarily cost efficient, as it results in a full warehouse only when a delivery has just taken place. A slightly smaller warehouse is likely to be more cost efficient.

Normal holding costs are set to 0.2 per period per part, 10% of the part cost. If we go past the limit of our warehouse, we will incur 0.4 per period per part over the limit. Calculating this limit falls outside the scope of this paper, as it deals with warehouse sizing. We will put this limit at an arbitrary 90% of the highest stock level of the 3 states. For most parameter sets this means we will incur increased costs for one or two periods after delivery. Delivery parameter set

2 has much higher probability of delivery, thus we expect this change to cause a larger increase in cost than for delivery parameter set 1.

4.2 Implementation

Given the 3 changes to costs, we are faced with 8 different scenarios, where the original paper had 4. All scenarios will include disruption risks and random lead times. We use the same method as Hekimoglu et al. (2018) to establish effects of the different properties: we determine optimal stock levels without taking certain properties into account, and then feed these stock levels to a simulation that does use them. The resulting differences between the base case, where all 3 mechanics are used in the stock level calculation, and the scenario give us the combined effect of one or more properties. Determining these stock levels requires us to modify the simulation from paragraph 3.1.2 to accommodate different prices for order and holding costs and to determine when priority delivery is useful. We will now explain these modifications in further detail.

Implementing different order costs for different states is a very simple modification, where we replace the original order cost of 2 with a vector with values [1.5 2 2.5]. The priority delivery will be activated once there is less than one period of expected demand in stock. If a priority delivery is activated, the probability of no delivery is decreased by 50% until the next delivery takes place. For example, if we have 3 states and the delivery probability vector [0.6 0.4 0.2], the complementary no-delivery probability vector will be [0.4 0.6 0.8]. We then halve each element of this vector to [0.2 0.3 0.4]. Then the probabilities of delivery change to [0.8 0.7 0.6]. Notice the absolute effect of priority delivery is stronger when the supplier is in an unhealthier state. However, in this state priority delivery also costs more, since base order cost is higher. Finally, we implement the increased holding costs by counting inventory above the 90% threshold twice.

The calculation of each individual effect is simple, but coupled effects need to be considered more carefully, since using the deviation of the scenario ignoring both effects compared to the base case also has the individual effects included. Therefore, we need to subtract the individual effects from this deviation. Using Δ^o, Δ^p and Δ^h for the effects of discounting order cost, allowing priority delivery and the two-stage holding cost function, Δ^{op} to indicate the combined effect of order cost and priority delivery and Δ^{op}_* for the deviation of the scenario ignoring both order cost and priority delivery, we calculate coupled effects using the following functions:

$$\Delta^{op} = \Delta^{op}_* - \Delta^o - \Delta^p \tag{7}$$

$$\Delta^{oh} = \Delta^{oh}_* - \Delta^o - \Delta^p \tag{8}$$

$$\Delta^{ph} = \Delta^{ph}_* - \Delta^p - \Delta^h \tag{9}$$

$$\Delta^{oph} = \Delta^{oph}_* - \Delta^{op} - \Delta^{oh} - \Delta^{ph} - \Delta^o - \Delta^p - \Delta^h \tag{10}$$

4.2.1 Theoretical analysis

The changes to pricing mechanics means the equations from 3.1.1 are no longer valid. However, we can update them to incorporate all 3 changes. The change in order costs is easily added, since order costs are fixed in Equation 3 as the state is given. Thus the only change is that instead of using c we use c_i. This does not affect any of the results about the function from Hekimoglu et al. (2018), such as its convexity in x.

Secondly, the 2-stage holding cost function requires a change to Equation 1, which changes to

$$C^l(x) = \alpha^l \mathbf{E}(h \max((x - D_{l+1}), 0) + p \max(D_{l+1} - x, 0) + h \max((x - D_{l+1} - z), 0)) \quad (11)$$

where z is the warehouse capacity. The function remains convex in x.

Accommodating priority delivery is more difficult, as it affects $Pr\{L(i) \leq l \leq L(i_+)\}$. However, the only function using this statement is $C(i, x)$. Since priority delivery is dependent on x only, we can extend this function using 2 sets of $Pr\{L(i) \leq l \leq L(i_+)\}$, one for normal delivery and one for priority delivery. We will call the priority lead time of state i $L^*(i)$. Then $C(i, x)$ changes to

$$\begin{aligned} C(i,x) = I(x \geq \mathbf{E}(D)) \sum_{l \geq 0} Pr\{L(i) \leq l \leq L(i^+)\} C^l(x) \\ + I(x < \mathbf{E}(D)) \sum_{l \geq 0} Pr\{L^*(i) \leq l \leq L^*(i^+)\} C^l(x) \end{aligned} \quad (12)$$

where I is the indicator function. This function also remains convex in x, since both of its parts are convex in x. It can also easily accommodate higher or lower thresholds for priority delivery by replacing $\mathbf{E}(D)$.

A similar change is required for Equation 3, as priority delivery affects order cost. We add the following statement to the end of the equation:

$$I(x < \mathbf{E}(D)) cy(1 - \alpha q(i)) \quad (13)$$

This statement is linear in y and convex in x.

Since each change has been added to the functions governing base stock levels and none of the assumptions required to prove that a state-dependent base stock level policy is optimal have changed, it follows that a state-dependent base stock level policy remains optimal.

5 Results: replication

In this section, we compare our results to those of Hekimoglu et al. (2018). We start with comparing the effects of random lead times and disruption risk as explained in Paragraph 3.1.4. Then we report on differences between stock levels. Finally, we present the simulation results for the business case presented in Section 5.5 of Hekimoglu et al. (2018).

5.1 Cost comparison

We have chosen to present results in the same manner as Hekimoglu et al. (2018) to make the comparison easier. Below we show the relative cost deviations.

First of all, we note the difference in disruption effect between 5% and higher percentage of disruption periods. It is very low when there are few disruptions, but reaches near 40% for the SFD-unstable case. While the patterns are not the same as the paper, the effect of disruption is very similar in the LID case. Lastly, we see a much larger effect of disruptions in the SFD case than in the paper.

The effect of random lead time varies wildly, being slightly (-8%) negative for the first delivery parameter set, while it is positive for most cases. Its effect is also very obvious on LIDs with the second delivery parameter set, which is very different from Hekimoglu et al. (2018). We do spot the same trend of its effect decreasing as the system spends more time in disruption.

The combined effect varies less than the effect of random lead time, having -1% as its minimum but 23% as its maximum. It is mostly positive and its effect increases as the percentage of disruption periods increases. In general, the combined effect appears smaller than the results from Hekimoglu et al. (2018) suggest.

Furthermore, the effect of nonstationarity is quite small, having no to a slightly negative effect. Once again, it is much smaller than the paper, whose results indicate an effect between 15 to 35%. An increase in the percentage of disruption periods appears to have little effect. However, the relative differences are very small, which combined with an non-rejected two-sample t-test leads us to believe nonstationarity does not have an effect on costs.

Finally, we note that total effects are much smaller than found in Hekimoglu et al. (2018) and in certain cases it may be more efficient to ignore these properties in favour of expanding the model in other directions.

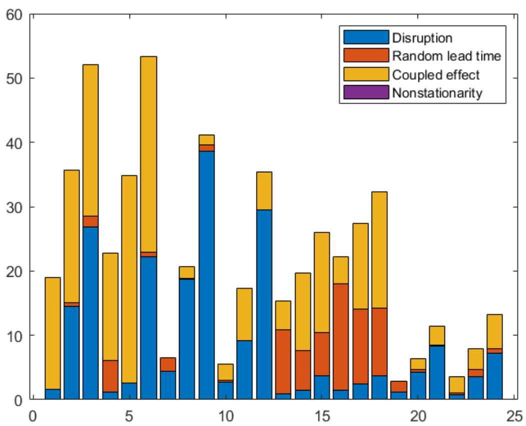

Figure 2: Positive cost deviations. From left to right (each 3 bars): stable LID, unstable LID, stable SFD, unstable SFD, repeated for delivery parameter set 2

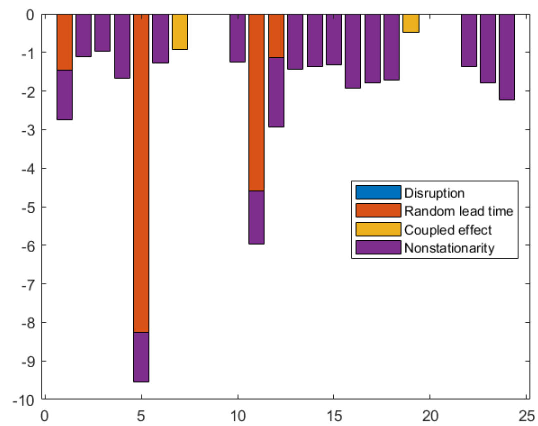

Figure 3: Negative cost deviations. Order is the same as in the figure to the left

Explaining these differences is difficult. Small differences are to be expected when using simulation compared to a theoretical analysis, but our results are sufficiently different we must look for other explanations. It is also difficult to compare absolute cost deviations since Hekimoglu et al. (2018) did not provide them. We do however have one set of stock levels for the LID-unstable case: 11, 18, 26 for the 3 healthy states. In the next section we compare this to the optimal stock levels we found and use this to explain the large differences in cost deviations found in this section.

5.2 Stock levels

We calculated optimal stock levels using a relatively small number of simulation iterations, while the paper used the value iteration method to determine these in an analytic manner. This means small differences will be due to the use of simulation and not because of any problems in the methods used. Stock levels for the base case are given in Table 1.

Parameter set	1	2	3	4	5	6	7	8	9	10	11	12
state 1	14	15	14	15	14	15	14	16	14	17	15	17
state 2	19	20	18	19	24	22	17	19	25	24	19	18
state 3	21	21	22	24	45	21	22	23	55	49	24	24
Parameter set	13	14	15	16	17	18	19	20	21	22	23	24
state 1	9	9	8	9	9	9	9	9	9	10	9	9
state 2	10	9	9	9	18	11	10	10	19	19	10	10
state 3	16	16	11	11	39	15	13	13	47	43	15	14

Table 1: Stock levels for all 24 parameter sets, base case. Order follows that of the (second) parameter table in Appendix A

We know of only one set of stock levels in Hekimoglu et al. (2018), which is [11,18,26] for the LID-unstable case. The percentage of disruption periods is unknown. We do not find any equal stock levels, but there are scenarios that have comparable stock levels. We do not observe non-decreasing stock levels in the state, as on occasion the third state has a slightly lower stock level than the second state. We believe this has to do with the small number of simulation iterations used to determine optimal stock levels. The variance of the costs varies between 10000 and 2.6 million between scenarios, which means a large sample mean variance of 100 to 25 000 remains even when using 100 iterations. The standard deviation of the sample mean then lies between 10 and 140. However, for most cases the base variance is lower than 80 000, which leads to a standard deviation of the sample mean of approximately 30. This means we are unlikely to select the optimal stock level, but because average costs are approximately 750 we will not be too far from the optimal cost.

This large variance is present as well in the final cost calculation, but since we use a much larger number of iterations its effect is much smaller. We hypothesise that a small number of iterations may spend an extremely long time in a disruption state, causing much larger costs than usual. This view is supported by the high variances found in scenarios with 15% disruption periods and LIDs.

The lack of absolute cost results from Hekimoglu et al. (2018) makes it difficult to determine whether our relative cost deviation are smaller because the effect is actually smaller than found by the paper or if we overestimated base case cost while underestimating other scenario's. However, when the results of scientific research cannot be checked, it is by definition invalid. Absolute costs can therefore be found in Appendix B for the future researcher.

In total we find smaller effects for all properties but are unable to attribute it to anything except the use of simulation due to the lack of absolute costs in the results from Hekimoglu et al. (2018).

5.3 Business case

The paper finds 2 sets of stock levels: [9,15] for 90% service rate and [16,30] for 99% service rate. We find much higher values of [16,29] and [26,43]. It is unlikely to be a result from using simulation, although it is strange that what we find to be 90% service rate stock levels is almost equal to the papers 99% stock level. However, it is possible the paper used a different value for expected demand, which makes any comparison of stock levels meaningless. Assuming they continued using an expected demand of 2, however, we must attribute the increase in stock levels to the use of simulation to determine the stock levels. We find much smaller sample mean standard deviation than in the basic problem, but it is still equal to the increase in cost expected if both stock levels were to increase by 7, which is $100 * 0.1 * c$, or 10 times the order cost (assuming no shortages occur). This means an decrease in the stock level of 14 is at the limit of a one-sided 95% confidence region, assuming a symmetrical distribution of stock levels around the optimal stock level. Thus we do not reject the results from Hekimoglu et al. (2018).

6 Results: extension

We have summarised the positive effects of each change in pricing mechanics in Figure 4.

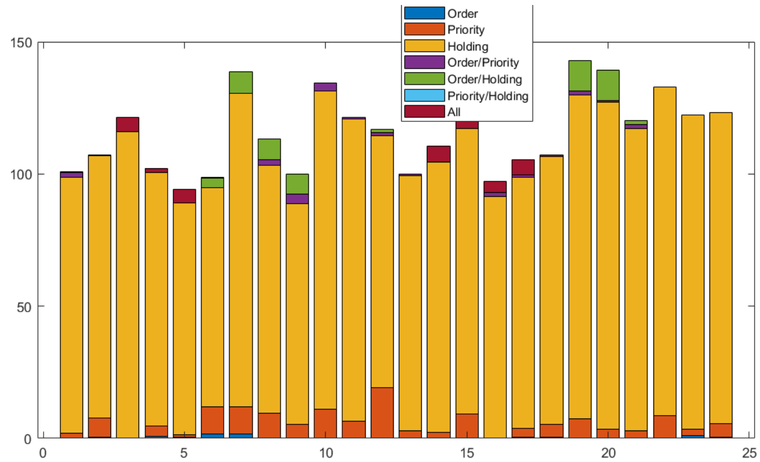

Figure 4: Relative cost deviations for each set of effects. From left to right (each 3 bars): stable LID, unstable LID, stable SFD, unstable SFD. Repeat for delivery parameter set 2.

First of all, we note that linking order cost to supplier health has little effect on its own, but significantly increases cost when combined with other factors. These results invalidate our hypothesis about priority delivery; it has a larger effect in the case of SFDs compared to LIDs. The 2-stage holding cost function has a very large effect on all cases, but a larger effect on SFDs, as predicted. The stock levels associated with this scenario are extremely low (below 5 for all cases), which means we are likely overestimating this effect. We also see how its effect decreases as the percentage of disruption periods increase, because inventory is unlikely to be above the 90% limit during a disruption. This is false for the LID-stable case, where the effect increases instead.

Secondly, the combined effect of order cost and priority delivery is a slight increase in costs of 1-3%, because the change in order costs also increases the cost of priority delivery. Finally we see the small combined effect of order and holding cost changes in both SFD-stable cases, which is likely the result of the stock level smoothing that the order cost changes create. Smoother stock levels result in a lower warehouse capacity, which means we exceed it more often. The combined effect of all changes is quite small and only noticeable in the LID cases.

Now we move on to the negative effects on costs, shown in Figure 5.

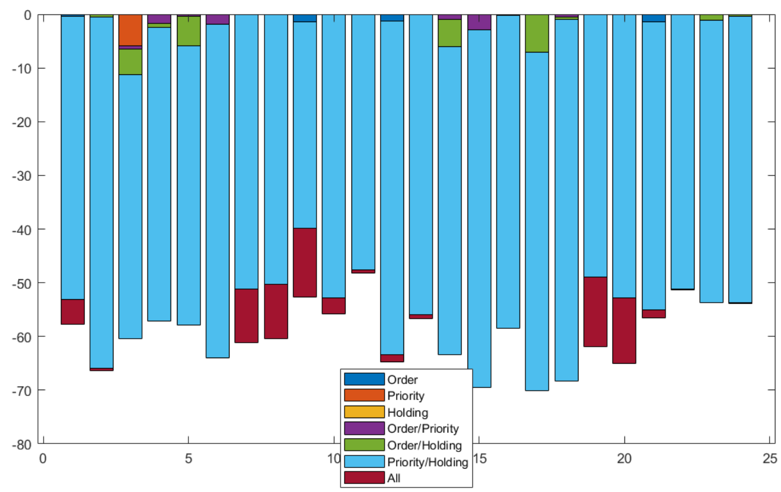

Figure 5: Relative cost deviations for each set of effects. From left to right (each 3 bars): stable LID, unstable LID, stable SFD, unstable SFD. Repeat for delivery parameter set 2.

We see small insignificant negative deviations due to order costs, and one occasion where ignoring priority delivery results in lower total costs. This effect is still very small and happens only once, which means it is difficult to attribute it to a cause.

The combined effect of priority delivery and the two-stage holding cost is strongly negative (40% - 70%). We attribute this to the large positive effect of the two-stage holding cost function, which may be invalid as it chooses unrealistically low stock levels (below 5 in all cases). There may be a logical error involved, which would make all results involving the two-stage holding cost function invalid.

Summarising our results, we find small positive deviations for most changes and their combined effects, and very large deviations for the two-stage holding cost function. These large deviations are likely the result of a logical error in our program.

7 Conclusion

We start with summarising our findings in section 7.1, followed by detailing the implications of our research in section 7.2. Finally we explain parts of the problem that could use additional research in section 7.3.

7.1 Summary

We describe a non-stationary inventory system with disruptions and random lead times. We replicate the work of Hekimoglu et al. (2018) and find comparable results as most differences are a result of simulation. Secondly, we extend this inventory system with 3 changes to cost parameters related to order costs, delivery and holding costs.

We follow up with determining relative cost deviations of each set of mechanics and accommodating functions from Hekimoglu et al. (2018) to incorporate these mechanics. We find that these changes do not violate any assumptions required for the optimality of a state-dependent base stock policy. Our results indicate small positive deviations for each of our changes in cost parameters.

Our results are not close enough to Hekimoglu et al. (2018) to validate their work, however due to lack of absolute costs in their results we cannot determine whether our work or theirs has made errors.

7.2 Implications

We find that in certain parameter sets the total cost deviation of disruption, random lead times and non-stationarity are below 10%. Models should always attempt to reflect only relevant mechanics of the problem, which means that in some cases these properties can be ignored at only a small increase in cost.

Secondly, we find that the efficiency of an inventory system can be improved by adding priority delivery, even when the cost of such a delivery is very high.

Thirdly, it is apparent that estimating stock levels using simulation is very inefficient, both in computation time and quality of results, and should be done using value iteration whenever possible.

Finally, we note that making order costs dependent on the supplier state smooths stock levels, which is desirable for a supplier using a make-to-order production system. This creates more continuous production levels and prevents periods where no production occurs due to a lack of demand.

7.3 Future research

First of all, each of our changes in price assumed some arbitrary threshold or value, 25% for the discounting, priority delivery once inventory went below expected demand and warehouse capacity of 90% of the maximum base stock level. We were unable to do a sensitivity analysis due to computational complexity, but it will certainly be interesting to see what optimal thresholds would be. We advise using the value-iteration method to determine stock levels as we have already incorporated these mechanics into the cost functions and it is much faster.

Secondly, we accommodated the cost equations for the changes in pricing and proved an optimal base stock policy remained optimal. While the theory is valid, such results always need practical validation, which we have not given.

Lastly, we had other ideas to extend the research of Hekimoglu et al. (2018), those being a multi-part inventory system with repairs requiring more than one type of part and relaxing the assumption of perfect knowledge of the supplier state. Some research into these extensions is quite necessary as the assumptions of perfect knowledge and single-part repairs are true only in a small number of cases.

References

P. Alfredsson and J. Verrijdt. Modeling Emergency Supply Flexibility in a Two-Echolon Inventory System. *Management Science*, 45(10):1416, 1999.

M. Hekimoglu, E. van der Laan, and R. Dekker. Markov-modulated analysis of a spare parts system with random lead times and disruption risks. *European Journal of Operational Research*, 269(1):909–922, 2018.

B. Lev and H. Weiss. Inventory Models With Cost Changes. *Operations Research*, 38(1):53, 1990.

A Addendum

We have made some changes to the parameter sets given in the online supplement of Hekimoglu et al. (2018). Figure A shows the original table, while the second figure shows our table. We made 2 changes: first of all, e should be 0.08 in the stable cases, and the indicated order of LID's and SFD's is wrong, as can be seen when looking at disruption and recovery probabilities.

5% Disruption Periods	e	d	$q(1)$	$q(2)$	$q(3)$	$\zeta(d_1)$	$\zeta(d2)$	$\zeta(d3)$
stable-LID	0.8	0.1	0.001	0.0105	0.014	0.256	0.128	0.085
stable-SFD	0.8	0.1	0.001	0.0075	0.01	0.248	0.124	0.083
unstalbe-LID	0.12	0.1	0.001	0.0375	0.05	0.99	0.495	0.33
unstable-SFD	0.12	0.1	0.001	0.0255	0.034	0.95	0.475	0.317
10% Disruption Periods								
stable-LID	0.8	0.1	0.001	0.0195	0.026	0.216	0.108	0.072
stable-SFD	0.8	0.1	0.001	0.0135	0.018	0.204	0.102	0.068
unstalbe-LID	0.12	0.1	0.001	0.0765	0.102	0.98	0.49	0.327
unstable-SFD	0.12	0.1	0.001	0.0555	0.074	0.96	0.48	0.32
15% Disruption Periods								
stable-LID	0.8	0.1	0.001	0.0285	0.038	0.192	0.096	0.064
stable-SFD	0.8	0.1	0.001	0.0225	0.03	0.212	0.106	0.071
unstalbe-LID	0.12	0.1	0.001	0.1245	0.166	0.99	0.495	0.33
unstable-SFD	0.12	0.1	0.001	0.0915	0.122	0.99	0.495	0.33

5% Disruption Periods	e	d	$q(1)$	$q(2)$	$q(3)$	$\zeta(d_1)$	$\zeta(d2)$	$\zeta(d3)$
stable-LID	0.08	0.1	0.001	0.0105	0.014	0.256	0.128	0.085
unstable-LID	0.12	0.1	0.001	0.0075	0.01	0.248	0.124	0.083
stable-SFD	0.08	0.1	0.001	0.0375	0.05	0.99	0.495	0.33
unstable-SFD	0.12	0.1	0.001	0.0255	0.034	0.95	0.475	0.317
10% Disruption Periods								
stable-LID	0.08	0.1	0.001	0.0195	0.026	0.216	0.108	0.072
unstable-LID	0.12	0.1	0.001	0.0135	0.018	0.204	0.102	0.068
stable-SFD	0.08	0.1	0.001	0.0765	0.102	0.98	0.49	0.327
unstable-SFD	0.12	0.1	0.001	0.0555	0.074	0.96	0.48	0.32
15% Disruption Periods								
stable-LID	0.08	0.1	0.001	0.0285	0.038	0.192	0.096	0.064
unstable-LID	0.12	0.1	0.001	0.0225	0.03	0.212	0.106	0.071
stable-SFD	0.08	0.1	0.001	0.1245	0.166	0.99	0.495	0.33
unstable-SFD	0.12	0.1	0.001	0.0915	0.122	0.99	0.495	0.33

We continue with the 2 delivery parameter sets. The first is [0.6 0.4 0.2] and the second is [0.9 0.85 0.8]. The second set is used to determine the effect of random lead times when the first two moments of the distribution are near zero.

B Costs

Now we show optimal costs found in our simulations. The order of parameters is as given in the tables above with [0.6 0.4 0.2] as delivery parameters for the first 12 scenarios, then [0.9 0.85 0.8] for the next 12.

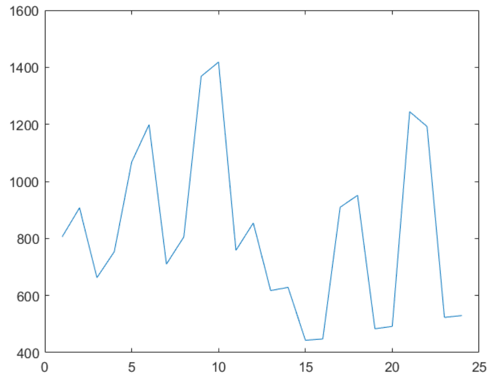

Figure 6: costs ignoring random lead time and disruption

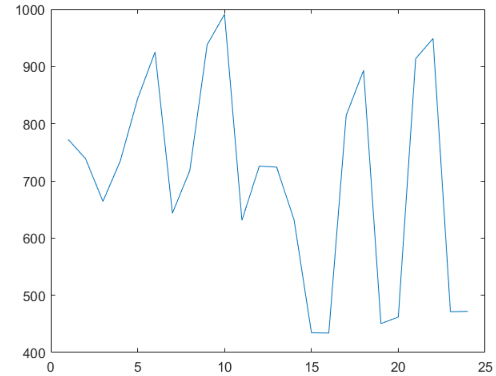

Figure 7: costs ignoring disruption

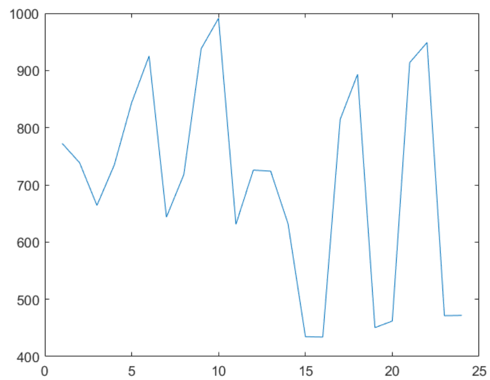

Figure 8: costs ignoring random lead time

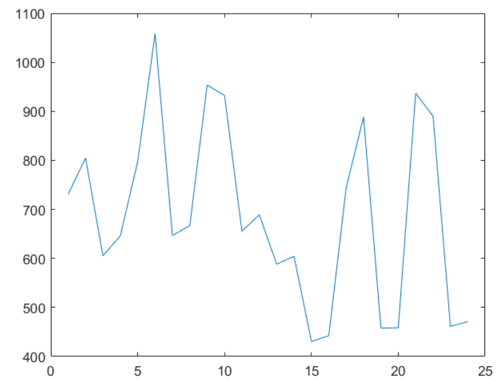

Figure 9: costs base case

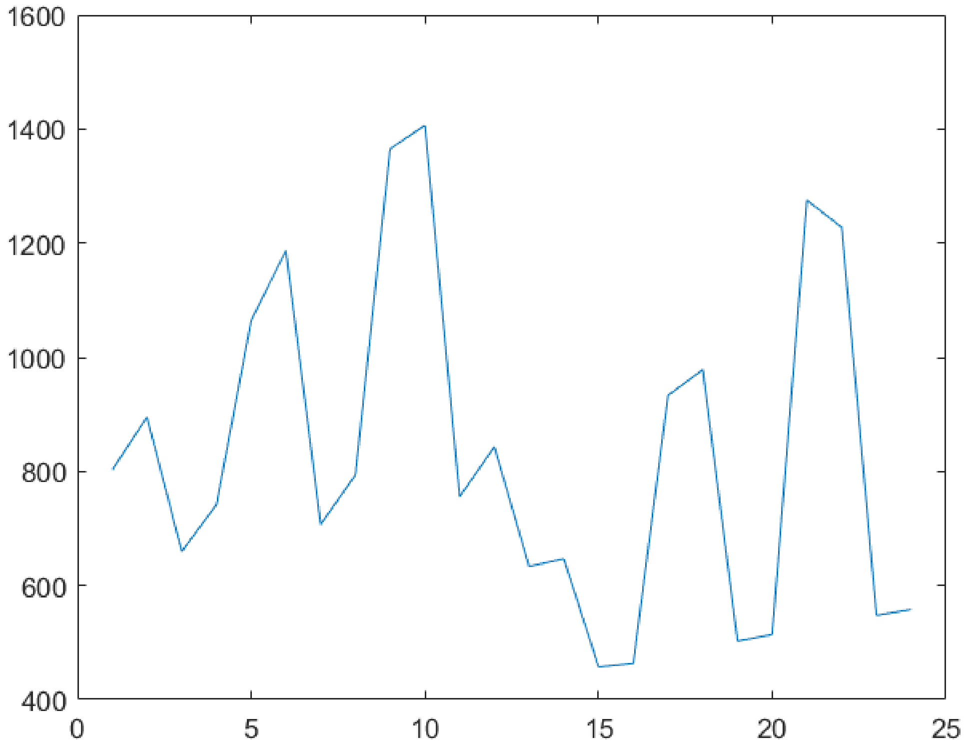

Figure 10: costs ignoring random lead time, disruption and non-stationarity

C Code

As we have written over 30 functions, adding them here is too much. They are included in a zip file. We will explain each function briefly here

Program name	Function
Algorithm 1	Implements Algorithm 1 from Hekimoglu et al. (2018)
B_GetResults	Main executive function for the business case
B_Simulation	Simulation tailored to business case, mostly returning the service level
C_ix	Implements Equation 2
C_l_x	Implements Equation 1
CalculateMarkovMatrix	Calculates the transition matrix from the given probabilities
Costs	Cost function for a single period in simulation for replication
Deliver	Delivers orders
Deliver_Det	Delivers order when lead times are deterministic
Demand_Subtraction	Calculates a single discrete Poisson variable using Knuth's algorithm
E_Costs	Cost function for a single period in simulation for extension
E_CreateBarPlots	Calculates equations 7
E_Deliver	Delivers orders, for extension
E_GetAllResults	Main executive function for extension
E_Run	Sub-executive function, calculates stock levels and then runs simulation 50 000 times
E_Simulation	Runs a single simulation, for extension
E_StockLevels	Calculates optimal stock levels using enumeration
E_StockLevelsEfficient	Calculates optimal stock levels after E_StockLevels found initial result
G_Cap_n_iy	Recursive function, implements 3
g_n_iy	Recursive function, implements 4
g_new	Recursive function, implements 4 but is only recursive in g_new.
GetAllResults	Main executive function for replication
GetDetLead	Calculates deterministic lead times from the probabilities Algorithm 1 finds
Order_Placement	Places an order in a simulation
Order_Placement_Det	Places an order given deterministic lead times in a simulation
R_CreateBarPlots	Same function as E_CreateBarPlots but for replication
Run	Sub-executive function for replication part, calculates stock levels then runs the simulation 50 000 times
Simulation	Runs a single simulation for replication
StockLevels	Calculates optimal stock levels for replication using enumeration
StockLevelsEfficient	Calculates optimal stock levels after StockLevels found initial result
Supply_Change	Changes the state of the supplier using the transition matrix
temp	Separates positive from negative values, used to creates cost deviation figures
Value_Iteration	Executes the value iteration method
results.mat	Contains all final results

Table 2: Short explanation of each function, in alphabetical order