ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Bachelor Thesis Quantitative Logistics

Spare parts system analysis with supply risks and a recycling option

Wietze Janssen 455018

July 7, 2019

Supervisor: Ma, W Second assessor: Dekker, R

Abstract

For a supplying company it is important to have enough parts on stock to satisfy their customers. While finding the optimal stock level the company has to take into account a lot of different factors, especially when the company is depending on another supplier company that has to deal with supplying risks as random lead times and supply disruptions. We will look at the effects of not considering the risks in the optimal policy by using a markov-modulated queuing system created by Hekimoğlu et al. (2018). We also will try to reduce the total costs by adding an option where the orders can be fulfilled by recycled parts. For the company the most important thing is knowing the situation they need to deal with, no matter how hard the situation is you have to know so you can adapt to the situation and get the best results possible.

1 Introduction

For every company it is important to be able to use the machines and the associated spare parts. When the lifespan of capital goods from a company come to an end, because the high value of capital goods, the company may still want to keep the capital goods and try to make the most out of it. In this case they are dependent on Original Equipment Manufacturers (OEMs) for spare parts supply. However OEMs are dealing with supply-side risks, for example, changes in technology (Rojo et al., 2010), transportation and logistic issues, lack of profitability, competition, financial problems and bankruptcy of suppliers (Babich et al., 2007). Hekimoğlu (2015) and Li et al. (2016) show that supply disruption risk is coupled with lead time variation. For an OEM it is important to have enough spare parts on stock, to fulfill the orders they receive within a certain service level. Because when the OEM receives an order and they do not have the spare parts on stock they can lose a certain amount of income. On the other hand when they have too many spare parts on stock the costs will go up because of the holding costs. An OEM is dealing with random lead times, possibilities of supply disruptions and transportation failures for the components of their products. So for an OEM it can be harmful to their profit if they do not know how to handle random lead times and possibilities of supply disruptions. Also we are looking into the case that the OEM has a maximum number on replenishment orders, in this way they can not make huge orders to compensate the loss of previous periods. Therefore the motivation for this research is to create a procedure for an OEM to keep the costs as low as possible while satisfying the service level they set up. Next to the practical reasons for companies, for example, it is also interesting to see what the effects of supply risks have on inventory systems, for example the base stock level and the time when replenishment orders need to be made. For capital goods it is also very common to use second-hand/repaired spare parts, as some of the spare parts can be very costly. We want to see what would happen in our case if we would use a second supplier providing those second-hand/repaired spare parts, and what the effects are on the base stock level. To see the effects we will compare the results of the model with the recycling option and that of the model without the extra second-hand spare parts supplier.

The main goal of this paper is creating a procedure considering supply risks for spare parts to reduce the costs and make the inventory system associate better with disruptions. We will also see the effects on the costs and other performance measures by the different supply risks. With a second supplier that provides recycled parts we will find out how the costs and the effects of the supply risks will change. First we will start with formulating a cost model for the inventory of spare parts, we combine this cost model with random lead times and supply disruption risk created by a Markov chain. To create these Markov-modulated random lead times, without crossovers, we will use a new queuing system. A crossover is when you have two orders and the order that is placed last, will be delivered first. With this queuing system we can also calculate the distribution of state-dependent random lead times. After this we will add the recycling part, the transportation failures and a maximum on the replenishment orders separately to see what happens to the performance measures. Our main findings are that when the OEM do not take certain supply risks into account while they have to deal with them, they can end up with high unexpected costs. And introducing the option of a second-hand spare part supplier decreases the costs and the effects of supply risks. The remaining of this research is structured as follows. We compare our research to the literature in Section 2. And we will discuss the methods we used in Section 3. Then we will show our results and conclusion in Sections 4 and 5. Finally we will discuss our research in Section 6.

2 Literature

Our research is mainly based on the paper of Hekimoğlu et al. (2018). In line with Hekimoğlu et al. (2018), our paper has two topics, the first of which is random lead times. We assume that outstanding orders cannot cross each other in the supply system, addressed by Kaplan (1970). Kaplan (1970) showed that with the non-crossover lead times, multidimensional minimization problem can be reduced to a one-dimensional minimization problem. Additions to this model made by Ehrhardt (1984) and Song and Zipkin (1996) lead to optimal base stock policies and Markov-modulated random lead times but without supply disruptions. In our paper we consider random lead times without crossover, but interesting studies that allow crossovers are, Robinson et al. (2001); Bradley and Robinson (2005); Disney et al. (2016); Hayya et al. (2011).

For the second topic supply disruptions we use two certain times, “up-time” and “downtime” (Tomlin, 2006). For this problem we use, as well as in (Özekici and Parlar, 1999), an exogenous Markov chain to model the random lead times. But a major assumption they make is immediate delivery of replenishment order, only in our case we are dealing with lead times that are random. Tomlin (2006) suggests two suppliers and an optimal policy taking into account the “up-time” and the nature of the disruptions. We will focus on the combined effects of random lead times and supply disruptions.

For the part where we include a second-hand supplier we used the paper of Lou et al. (2017). Where Lou et al. (2017) has done research into recycling parts for a lot of different products. With the results of this paper we can make an assumption for the percentage we can recycle from the used products. In Hekimoğlu (2015) they also included a second-hand supplier, and compared the results to the case without the second-hand supplier.

3 Methodology

In this study we try to create a procedure and analyze the optimal inventory management taking into account different supply risks. The methods, formulations and data we use are mainly based on (Hekimoğlu et al., 2018). We will extend their research a little bit further with introducing a second supplier providing recycled parts.

3.1 Mathematical notation

In our model we are dealing with a discrete time, single-item single-echelon, inventory system. The replenishment orders, are delivered with random lead times. To this problem we had to make the assumption that outstanding orders cannot cross each other in the supply process. This assumption is based on (Kaplan, 1970). This assumption can be to strict when your are dealing with a lot of different orders in a small time period, because when there are so much orders the chance of a crossover happening is higher. But we are dealing with spare parts which are slow moving, so this assumption will not create any problems in our case.

The events that occur every period are ordered as follows. First, the inventory decision maker knows the supply process and decides whether a replenishment order must take place this period or not. Hekimoğlu et al. (2018) assumes fixed ordering cost to be zero, hence only the acquisition costs (c, c_r) must be charged at the time of the order (D, D_r) placement. Then, previous orders are delivered, customer demand are realized and holding and shortage costs (h and p) are incurred. And finally the state of the supplier might changes. We use *L(i)* as a discrete random variable for lead time of an order when the supplier is in state i. And i_+ is a random variable indicating the next healthy state after state i, d^i is the random variable indicating the disruption state of healthy state i. And d^i_+ is the random variable indicating the next disruption state after the disruption state d_i.

3.2 Supply disruption risk

To take supply disruption risk into account, we use an exogenous, discrete-time Markov chain that controls the supply system. In the system we defined two sets. The first set B is the set of all possible states of the Markov chain. The second set B_h is a subset of B and includes all healthy states of the Markov chain. Only in healthy states replenishment orders can be placed. So in disruption states no orders can be placed until it goes back to a healthy state. And when the supplier goes to a disruption state all the outstanding replenishment orders from the OEM to the supplier get discarded, but differently than in Hekimoğlu et al. (2018), we assume the OEM cannot receive any orders as well. Later in the paper we will also look at the case where the OEM can receive orders while the supplier is in disruption. These results will be discussed in Section 4.4.1. It is possible that the lead times and the disruption probabilities are different for the healthy states.

For every healthy state i two events that can happen, the supplier can stay in a healthy state with probability *q(i)*, or a supply disruption happens and the supplier changes to the the belonging disruption state d^i with probability 1 - *q(i)* ($\bar{q}(i)$). Also for every disruption state d^i two events can happen, either the system stays in the same disruption state d^i with probability $\xi(d^i)$, or it jumps to the corresponding healthy state i with probability 1 - $\xi(d^i)$ ($\bar{\xi}(d^i)$).

The transition probability matrix $\mathcal{P}$ on the set B has the following structure:

$$\mathcal{P} = \begin{pmatrix} \mathcal{Q}P^H & (I - \mathcal{Q}) \\ (I - \Xi) & \Xi \end{pmatrix} \tag{1}$$

where,

$$\mathcal{Q}(i,i) = q(i),\ \mathcal{Q}(i,j) = 0,\ \forall\, i\ \neq j \in B^h,$$

$$\Xi(d^i, d^i) = \xi(d^i),\ \Xi(d^i, d^j) = 0,\ \forall d^i \neq d^j \in B - B^h,$$

and

$$P^h = \left\{ p_{ij}\ :\ \textstyle\sum_j p_{ij} = 1,\ \forall\, i, j \in B^h \right\}.$$

In $\mathcal{P}$ first N rows are healthy states and the second N rows are are for the disruption states. And the sub-matrix P^h includes the probabilities of transitions between healthy states.

3.3 Recycling machine parts

Every time an OEM gets an order, we assume the company placing the order, has parts that are broken or just wants to renew their machines. For some valuable spare parts the used parts can also be recycled and repaired by second hand spare part companies. Lou et al. (2017) has done research in recycling machine parts for a lot of different products. The average percent of recycle materials at the end of product life for all the products is around 50 percent. We make a conservative assumption that we can recycle 25% of the products that are broken or abandoned. This means that we need four parts to create one part, so when the stock level is zero the supplier has to wait for broken or abandoned products, while the supplier using new parts does not have these problems. But if we come to a point where we have more recycled parts, the parts are in general delivered faster and the costs of the orders are lower. The acquisition costs (c) from the normal supplier are higher than the acquisition costs (c_r) of the supplier using recycled parts. And the lead time of the supplier providing recycled parts has a standard lead time instead of a random lead time, the holding and shortage costs are staying the same. Because the costs are lower and the lead time is constant and lower on average we always order recycled parts if possible, but because the supplier using recycled parts can only use 25% of the used parts the capacity is limited. Another advantage is that when the "normal" supplier is in disruption the second-hand spare part supplier can still deliver, but only when the number of repaired parts satisfies the replenishment order. We assume that the used parts from the customers of the OEM are sent to the second-hand spare part supplier, directly after an order is placed to the OEM by a customer. In this way the period after the order is placed, the used parts can be taken into account in the recycled parts stock of the second-hand spare part supplier.

3.4 Transportation failures

Every time an order needs to be delivered, it has to be transported by train, plain, truck or another vehicle. Every order has a chance of being delivered at the wrong place, the products being damaged or an accident happening on the route. This depends on the route the products need to take, for example a route with multiple vehicles has more opportunities where something can go wrong, so the chance of a failure is higher. If such a failure happens the OEM knows it when the order should have been delivered and then the OEM has to make a new order. For the OEM this takes a lot of time and money. We want to know the effect on the costs if we include a 1, 5 and 10 percent chance of a transportation failure. To see the biggest effect we include these percentages in the case where in disruption of the supplier the OEM can still receive orders. In the situation of 15% expected disruption periods, unstable supply process and the LID type of disruptions, the total costs are the highest. So we will compare the costs when we use 0, 1, 5 and 10 percent for the chance of a failure in this situation.

3.5 Maximum on amount of replenishment orders

We see from Hekimoğlu et al. (2018) that the ready and fill rates are always pretty high, this can be due to the fact that when the state changes to healthy again the next replenishment order is so high the optimal stock level is directly reached again. But in reality there can be a maximum of order size due to the capacity of the supplier, or the transportation mode. We will investigate what would happen to the costs, service rates and the effects if we would put a maximum on the amount of the replenishment order. We will check this under the same circumstances as we checked the effects of transportation disruption (15% unstable-LID). We want to know what happens under different maximum amounts, so we will compare the results by a maximum amount of 7, 10 and 13.

3.6 Optimal policy by minimizing the costs

To obtain the optimal policy under certain circumstances, we try to minimize the total costs over the entire period. The cost function we obtain every period is as follows:

$$C(x) = h \cdot max(x, 0) + p \cdot max(-x, 0) + c \cdot D + c_r \cdot D_r. \tag{2}$$

We use this cost function in every case we study, and the structure to find the optimal policy is in all cases the same. Where x is the inventory level at the end of the period and D is the demand of that period fulfilled by the "normal" supplier and D_r is the demand of that period fulfilled by the second-hand spare parts supplier. In our case the holding costs (h) and the backlog costs (p) are both per item per period. The acquisition costs (c) are for every item that is ordered. The total costs are all the single period costs added together. The cost function we use is derived from the holding and backlog cost function from Hekimoğlu et al. (2018). When introducing the second supplier the function stays the same, the only difference is when the replenishment order is fulfilled by the new supplier the acquisition costs change to c_r.

3.6.1 Heuristic

To get the optimal policy with the lowest costs possible we created an algorithm. In Algorithm 1 we first check if every stock level can go down one, every time the costs are lower than the best costs the "best" values (base stock levels) and "best" costs get updated. When the costs are higher than the best costs, then every stock level goes up one from the starting point until the costs are higher than the best costs. After this, we check if we are in a local minimum by checking if going up five or going down five decreases the costs, if so we do the first part again with a new starting value. The found policy will be the starting point for the second step, where we want to decrease and increase every stock level for every state separately with the same method as before. For this we have a similar algorithm but we decrease them separately now, so first we check if the stock level of state 2 can go down by one until the costs are higher than the optimal costs, then we do the same for state 1 and finally for state 0. After this we check if the stock level of state 2 can go up by one until the costs are higher than the optimal costs, then we do the same again for state 1 and finally for state 0. In this case we do not check if we are in a local minimum, because we already did this in the algorithm before. In Algorithm 1 we see the first step of the algorithm the second step is similar, and is explained above.

3.7 Markov-modulated random lead times by a queuing system

For modelling random lead times and taking into account supply disruption we use two semi-dependent queues suggested by Hekimoğlu et al. (2018). The first one is a discrete-time Bernoulli queue, the items in the queue are the healthy states (i) of the Markov chain. After every period the probability of staying healthy is $q(i)$ and the probability of a disruption happening is 1 - $q(i)$. When we stay in a healthy state the probability of an extra item arriving in the queue is e and an item leaving from the queue is d, so we made the assumption that we only can move one state up or down every period. When we are in a disruption state items cannot leave or enter the queue. When we jump back to a healthy state the number of items is the same as in the latest healthy state. With this queuing system we can calculate the elements p_{ij} of the transition matrix P^h.

To satisfy the no crossover assumption we use another discrete time queue (Zipkin, 1986), only we use *partial-batch bulk* service by this queue. In our case this means that in every period the batch (size K) get served at once with probability $b(i)$, or with probability 1 - $b(i)$ all items have to wait. In each period an item arrives at the queue with probability a until the capacity C is reached. When the capacity is reached the next incoming orders will be stored and processed when the current batch is delivered. In our case we assume the batch size (K) is equal to the capacity (C). An item that arrives is connected to that periods replenishment order, the amount of this order is equal to the base stock of the current state, minus the current inventory level, minus the

Algorithm 1 Algorithm to calculate optimal stock levels.

Optimal costs = 100000

Starting point = [5, 10, 15]

Current stock level = [5, 10, 15]

Optimal stock level = [5, 10, 15]

while Stop condition = true **do**

  Stop condition = false

  Check costs of current stock level

  if Costs < optimal costs & current levels are not increased yet **then**

    All current values decreased by one

    Optimal costs and optimal values get updated

  else if Costs ≥ optimal costs & current levels are not increased yet **then**

    New values are starting values increased by one

  else if Costs < optimal costs & current levels are increased yet **then**

    All current values increased by one

    Optimal costs and optimal values get updated

  end

  if Costs ≥ optimal costs & current levels are increased & current levels are not decreased by 5 **then**

    Set new values: best values decreased by 5

    Set starting values: best values decreased by 5

    Set already increased back to false

  else if Costs ≥ optimal costs & current levels are increased & current levels are decreased by 5 but not increased by 5 **then**

    Set new values: best values increased by 5

    Set starting values: best values increased by 5

    Set already increased back to false

  else

    Stop condition is true

  end

end

sum of the replenishment orders already in the queue. We do this to make sure that when the replenishment orders get delivered the inventory level is back on the base stock level of the current state. When the supplier is in a disruption we assume an OEM or another similar company cannot receive any orders as well. And all the outstanding replenishment orders get discarded, but in the next healthy state the replenishment order will be high enough to compensate the loss.

3.8 Inventory coverage algorithm

To calculate the probability $Pr\{L(i) \leq l \leq L(i_+)\}$ we use an algorithm proposed by Hekimoğlu et al. (2018). With this probabilities we can calculate the average leading time which we use for the case with deterministic leading times. In the algorithm we use a transition diagram A_1^i, which is denoted as follows:

$$\forall i \in B, A_1^i = \begin{pmatrix} 0 & 1-b(i) & b(i) \\ 0 & 1-b(i) & b(i) \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

When $(A_{l+1}^i)_{(1,3)}$ is used in the algorithm, it means the element on the first row and the third column of the transition matrix A_{l+1}^i.

Algorithm 2 Algorithm to calculate inventory coverage.

$A_1^{i+} = \sum_{j \in B} p_{ij} A_1^j$,

for all $l \geq 1$ **do**

  for all $i \in B$ **do**

    $A_l^{i+} = \sum_{j \in B} p_{ij} A_l^j$

    $A_{l+1}^{i+} := A_1^i A_l^{i+}$

    $Pr\{L(i) \leq l\} = (A_{l+1}^i)_{(1,3)}$

    $Pr\{L(i_+) < l\} = (A_l^i)_{(1,3)}$

$Pr\{L(i) \leq l \leq L(i_+)\} = Pr\{L(i) \leq l\} - Pr\{L(i_+) < l\}$

4 Results

To observe the single and combined effects of random lead times and supply disruptions we calculate the costs and the performance under different circumstances. The parameters we use are based on

Hekimoğlu et al. (2018). When we consider the case with deterministic lead times we use the average expected value of the lead times. If we want to consider the case without supply disruptions we set the corresponding parameters to zero. Hekimoğlu et al. (2018) found two different types of disruptions, first long and infrequent disruptions (LID) and second short and frequent disruptions (SFD). To distinguish the two different types we use different parameter values. The last attribute of a case is having stable or unstable supply offers. We are dealing with a stable supply scenario when the departure rate is higher than the arrival rate, so the probability of moving down a state is higher than moving up. For the unstable scenario it is the other way around, so the probability of moving up is higher.

With all these possible circumstances we can create 16 different cases. With all these cases we can observe the outcomes of having random lead times or supply disruptions. An example of the parameters we used is given in Table 1.

4.1 Design to obtain the results

For all the different circumstances we will calculate the optimal policy for all cases, with or without supply disruptions and random lead times. After this we will see how the policies perform in the case of supply disruptions and random lead times. We will measure these performance with the total costs and the fill and ready rates. Where the fill rate is the fraction of the orders that can be satisfied directly from the current stock level, and the ready rate is the fraction of the time the stock level is positive. With these results we can see the effects of taking supply-risk and random lead times into account while making a policy. To calculate these performance measures of all these different scenarios we use a simulation model. When we are making our optimal policy we try to minimize the total costs, in our case the holding costs are equal to 0.2, the backlog costs are equal to 4 and the acquisition cost are 2 for every item that is ordered.

4.2 Transition matrix $\mathcal{P}$

For the parameters e, d, $q(i)$ and $\xi(di)$ we use the parameter values from Hekimoğlu (2015). For the specific transition matrix $\mathcal{P}$ of the case with a stable supply process and 5% expected LID disruption periods, the parameter values we used are as follows:

With these values the transition matrix $\mathcal{P}$ has the following outcome:

Table 1: Parameter values specific example

e	d	q(0)	q(1)	q(2)	$\xi(\bar{d}^0)$	$\xi(\bar{d}^1)$	$\xi(\bar{d}^2)$
0.08	0.12	0.999	0.9895	0.986	0.256	0.128	0.085

$$\mathcal{P} = \begin{pmatrix} 0.999 \cdot 0.9198 & 0.999 \cdot 0.0802 & 0.999 \cdot 0 & 0.001 & 0 & 0 \\ 0.9895 \cdot 0.0921 & 0.9895 \cdot 0.8358 & 0.9895 \cdot 0.0721 & 0 & 0.0105 & 0 \\ 0.986 \cdot 0 & 0.986 \cdot 0.0922 & 0.986 \cdot 0.9078 & 0 & 0 & 0.014 \\ 0.256 & 0 & 0 & 0.744 & 0 & 0 \\ 0 & 0.128 & 0 & 0 & 0.872 & 0 \\ 0 & 0 & 0.085 & 0 & 0 & 0.915 \end{pmatrix}$$

In the first three rows we are currently in a healthy state and in the last three rows we are currently in a disruption state. In the first three columns the next state is healthy and in the last three columns the next state is unhealthy (disruption state). The values of the top left 3 x 3 matrix are the $\mathcal{P}_h$ values multiplied by *q(i)*. To calculate the values of $\mathcal{P}_h$ we used a time horizon of 10000 periods, because with 100 periods the probabilities from state two were less accurate, and we used 10000 replications to get the probabilities.

4.3 Inventory coverage

In Section 3.8 the Algorithm to calculate the probability $Pr\{L(i) \leq l \leq L(i_+)\}$ is given. It gives the probability of this periods order being delivered l-periods later. For the states 0, 1 and 2 the probabilities of the orders being delivered in 1, 2, 3, 4 or 5 periods are given in Table 2. These probabilities are calculated with the following values *b(0)* = 0.6, *b(1)* = 0.4 and *b(2)* = 0.2. From the probabilities we calculated for the first hundred periods, we conclude that in state 0 the average lead time is two periods, for state 1 this is three periods and for state 2 five periods. Those averages we used in the case where the lead times are deterministic.

4.4 Effects of supply disruptions and random lead times

In this section we will show the optimal policies of all the different cases in the situation of a stable supply scenario and long and infrequent disruptions (LID). The expected disruptions period are 5% of the total time horizon. After this we will also look at the cases with an unstable supply scenario, short and frequent supply disruptions and different percentages for expected disruption

Table 2: Probabilities of being delivered in l-periods

	State 0	State 1	State 2
1 period	0.25	0.24	0.16
2 periods	0.11	0.14	0.12
3 periods	0.05	0.09	0.09
4 periods	0.02	0.06	0.07
5 periods	0.01	0.04	0.05

periods. For all those scenarios we will see what the effects of the supply risks are. We will observe the effects for example from random lead times, by running the optimal policy found for the case with deterministic lead times and supply disruptions under the final case (both supply disruptions and random lead times). The deviation of the obtained costs from that and the costs of the optimal policy performing in the corresponding case is the effect of ignoring random lead times. The parameters values we use are shown in Section 4.2, with these values we can obtain all the parameter values of the different cases. The costs we show in Table 3 are the costs of the policy performing in the corresponding circumstances and the costs of the policy performing under random lead times and supply disruptions (final case).

Table 3: Optimal policies of the case stable-LID with 5% disruption periods

Policy	State 0	State 1	State 2	Costs in same case	Costs in final case
Det. LT	8	11	16	528.83	726.99
Rand. LT	11	16	20	682.97	674.21
Det. LT & supply disrup.	8	12	16	513.65	719.46
Rand. LT & supply disrup.	12	17	21	671.03	671.03

In Table 3, we see that when the company does not take into account random lead times, while estimating the budget for a certain period the final costs will deviate around 40%. For the management it is important to estimate the costs under the same case that they need to deal with, otherwise they can have a substantial debt at the end of the period.

To analyze the single and coupled effects of random lead times and supply disruptions we calculate the results under the different circumstances with different expected disruption periods. The parameter values we use are the parameter values Hekimoğlu et al. (2018) used as well. The

Table 4: Performance measures different policies in final case (stable-LID 5%)

Policy	Ready Rate	Fill Rate
Det. LT	0.91	0.89
Rand. LT	0.96	0.95
Det. LT & supply disrup.	0.92	0.90
Rand. LT & supply disrup.	0.97	0.95

results we found are shown in Figures 1, 2 and 3.

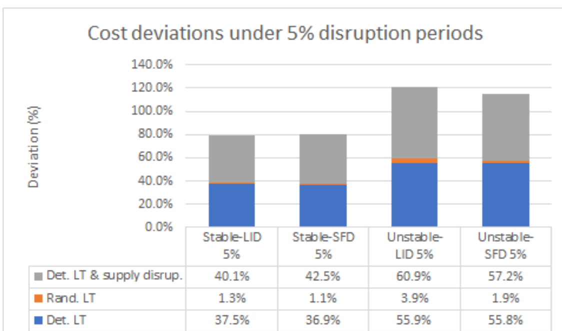

Figure 1

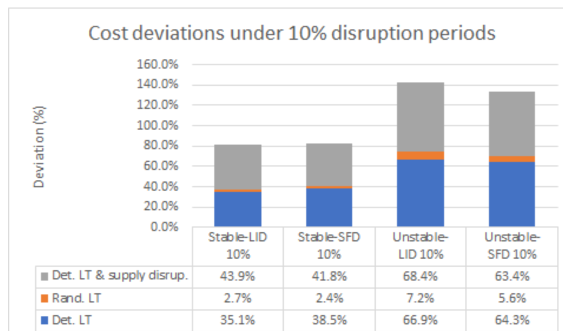

Figure 2

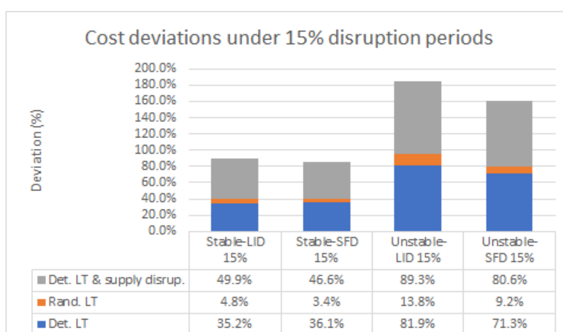

Figure 3

In the figures we see the policies in the table, so the effects are the other way around. For example the policy of only random lead times shows the effect of not taking into account supply disruptions. We see that under 15% expected disruption periods the effects are the highest in almost all cases. We see the effect of supply disruptions is the least significant and the effects of the random lead times has the biggest influence in the combined effects. This can be due to the assumption we made, that when the supplier is in a disruption an OEM or another similar company cannot receive any orders as well. It is even the case that without supply disruption the costs in the final case are lower so the deviation is negative, so when we look at the combined effect the negative deviation from supply disruption reduces the combined effects compared to the effect of random lead times. The deviation is negative because when we do not take disruption into account we can always make orders and the total acquisition costs will be a little bit higher and so are the total costs. And in Table 4 we see the that the fill and ready rate are always around 90% this can be due to the same reason. So to get a more clear view of the effects of supply disruptions, we will first run the same model with 15% expected disruption periods but without the assumption we made and after we will look at the results while using different values for *b(i)*.

4.4.1 Deeper research into supply disruption effects

In Figure 4 we see the cost deviation under 15% expected disruption periods for all the different cases, we do not use the assumption to get a more clear view of the effects of supply disruptions. Under the unstable supply process we see a cost deviation of 131.6% and 87.6% for LID and SFD respectively and in the stable situation we see cost deviations of 88.7% and 64.3%. If we compare the results of Figure 3 with Figure 4 we see that if the company does not follow the assumption the effects of the supply disruption are higher than following the assumption. In the unstable-LID case the difference is even 202.7 percent points. When comparing the policies under the two different cases we see that when we do not take disruption into account the policies are the same, but when we take disruption into account the base stock levels in the final case of Unstable-SFD 15% differ in every state with at least ten (from 15, 20, 24 to 25, 31, 36). So the optimal stock levels increased a lot and so did the total costs, when the OEM cannot receive any orders under disruption of the supplier the total costs are 761.24 and when the OEM can receive orders under disruption of the supplier the total costs went up to 1145.06. The big increase in deviation is because the difference between the optimal policies in different cases increased a lot. As mentioned before the optimal policies when taking disruption not into account stayed the same, but the optimal policy in the final case increased with at least ten. The deviation of 210.0% can be explained by the big difference in optimal policy that both have to perform in the final case, which leads to these high number of deviations.

In Figure 4 we also see that the combined effects are around twice as much as the biggest individual effects. And in both the stable and unstable supply process, the supply disruption effects have the biggest influence in the combined effects.

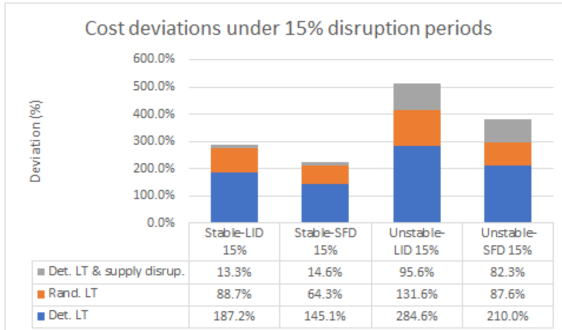

Figure 4

In Tables 5 and 6 we see the fill and ready rate of the unstable-LID case with 15% expected disruption periods with and without the possibility for the OEM of receiving orders in disruption periods. Both rates are always around 0.9 in the case following the assumption. In the case without the assumption the rates are more diversified and lower in every case. The rates are lower in Table 6 because the OEM can receive incoming orders when the supplier is in a disruption state, but the OEM themselves cannot make orders to the supplier so the inventory goes down further and the time of having a negative inventory level is higher. The same stands for the orders being served directly by the OEM, because this is only possible with a positive stock level. The rates being around 80/90% can be due to the fact that when the state changes to healthy again the next replenishment order is equal to all the outstanding replenishment orders. The time of having a really low inventory is relatively low because of this, and the rates are higher.

Table 5: Unstable-LID 15% (assumption)

Policy	Ready Rate	Fill Rate
Det. LT	0.85	0.84
Rand. LT	0.92	0.91
Det. LT & supply disrup.	0.85	0.84
Rand. LT & supply disrup.	0.96	0.94

When we use higher (0.9, 0.85, 0.8) values for $b(i)$ the chances of orders being delivered are higher. The results are shown in Figure 5. We see that the effects of random lead times are smaller, as wee can see the cost deviations are all below 10% while in Figure 3 the cost deviations were around 50 and 80 percent. This is because the uncertainty of orders being delivered is less, and

Table 6: Unstable-LID 15% (no assumption)

Policy	Ready Rate	Fill Rate
Det. LT	0.73	0.69
Rand. LT	0.82	0.78
Det. LT & supply disrup.	0.86	0.80
Rand. LT & supply disrup.	0.95	0.87

the frequency of deliveries is higher. The biggest difference with Figure 3 is the difference of the distribution of the effects, in Figure 3 the effect of random lead times plays the biggest role and in Figure 5 it is the effect of supply disruptions. Something else that is notable is that the combined effect of the cost deviation in Figure 5 is much lower. This can be explained by the fact that not taking disruption into account has a negative deviation effects, so the costs in the case with only random lead times are higher than the costs with random lead times and supply disruptions. This is because the chance of going to disruption is high in this case so the supplier will be much in the disruption state, when the supplier is in a disruption state the OEM cannot make orders so no costs are made. When we do not take disruption into account the supplier is always healthy so orders can be made at any time, the total acquisition costs are higher because of this and so are the total costs. So when we combine the "positive" deviation from random lead times with the "negative" deviation from supply disruptions the combined deviation will be smaller than the summation of the two individual effects.

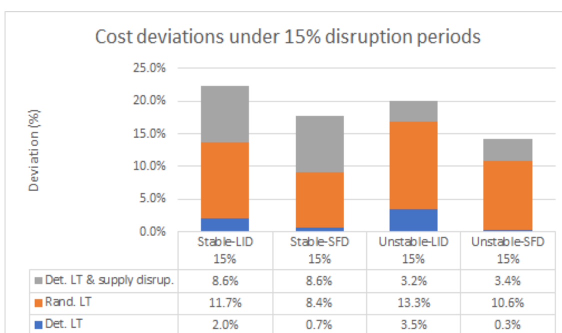

Figure 5

4.5 Recycling

When the option of using the second-hand supplier is introduced in the model, and the replenishment order is fulfilled by this supplier the acquisition costs (c_r) are 1.5 instead of 2 (c). And the deterministic lead time is three periods for every order fulfilled by the second-hand supplier, which is lower than the average lead time of the "normal" supplier in state and 2 and equal to the average lead time of state 1. In Figures 6 and 7 we see the results of including the recycling option into the model. In Figure 6 we see the effects have the same distribution as in Figure 4, with the effects of supply disruptions being the highest compared to the effects of random lead times. After all we see that in most cases the cost deviations are lower in Figure 6 than in Figure 4, so overall taking recycling into account reduces the effect in costs from supply disruption and random lead times.

In Figure 7 we see that the costs in every situation, under 15% expected disruption periods, are lower when we take recycling into account, with on average a cost reduction of 6.1%. So taking recycling into account reduces the costs and reduces the effect of supply disruptions and random lead times on the costs. When adding the recycling option the amount of the orders that are delivered by the supplier using recycled parts is always around 50%. So by changing 50% of the deliveries into recycled parts we reduce the costs with 6.1%. For the OEM this is positive, because it is a cost reduction, but for the "normal" supplier there are some negative effects. Most of the times an OEM delivers pretty exclusive parts and to just a few customers, but if you lose 50% of the orders to another supplier it reduces the income of the supplier by 50%. Adding the second-hand supplier might lead to a market failure, because the "normal" supplier will consider not supplying that particular product anymore and only a recycle supplier has not enough goods to maintain all the orders. If the "normal" supplier comes up with the recycling idea by themselves we will not confront this problem because it might even create more profit for the company.

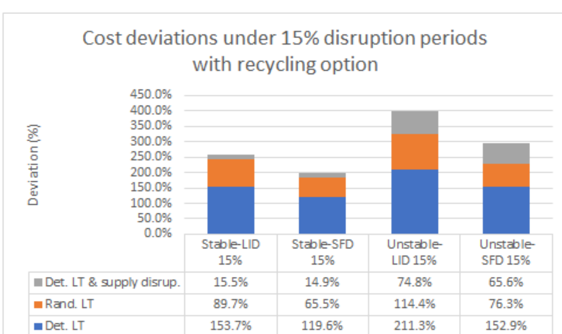

Figure 6

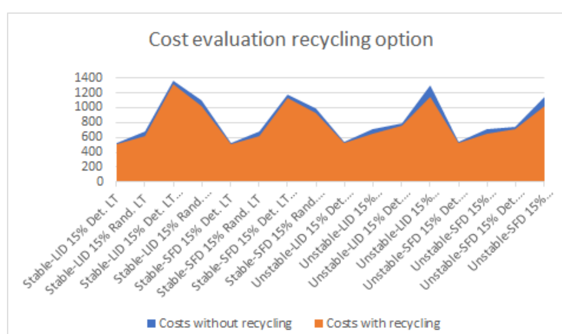

Figure 7

4.6 Transportation disruption

In Figure 8 we see that costs are increasing when the chance of a transportation disruption increases. The difference in costs until 10% disruption chance is below 3%, so even with 1 out of 10 orders not being delivered the costs barely goes up. Because the differences were this low we were wondering what would happen if the goods needed to be transported over one of the deadliest roads in the world the 'Yungas Road' in Bolivia, with over one hundred deaths a year Browne (2005), where the disruption chance might be around 50%. The costs are also shown in Figure 8. In this case the costs are around 25% higher than when the chance of transportation disruptions is zero. We can conclude when you do not need to travel over very deadly roads taking into account transportation disruption will not have a big influence on the costs. However if you know the transportation disruption chance you can always better take it into account, so you do not encounter unexpected costs.

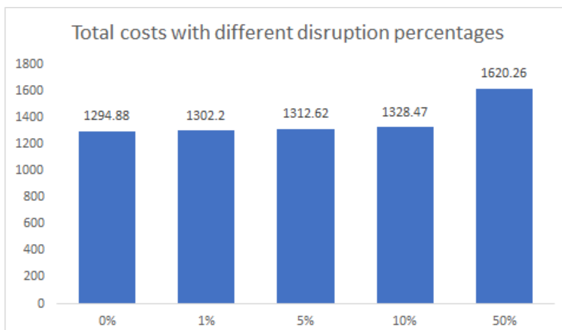

Figure 8

4.7 Maximum on amount of replenishment orders

In Figure 9 we show the effects of different maximums on replenishment orders. We see when the maximum of the replenishment order goes down the effects on the deviation goes up, especially the effects from supply disruptions and thereby the combined effects. In the situation with a maximum of 7 the deviation even is 517%, so if you do not take into account supply disruptions and random lead times while obtaining the optimal policy you will end up with costs five times higher. Mistakes like this can be fatal for companies, so the companies need to be aware of the situation they are in. In Figure 10 we see the costs of the optimal policy performing in the same situation, so for example the costs of the optimal policy with deterministic lead times performing in the situation with deterministic lead times. We see in the situation without supply disruptions the costs do not actually differ, but in the situation with supply disruptions we clearly see decreasing costs when the maximum amount goes up. The service rates of the optimal policies from the situation with only deterministic lead times are around 60% and the other service rates are still around 70/80%. So even when we have a maximum on the amount of the replenishment orders and we use a policy in the final case that changes a lot from the optimal policy we still have around 70% of the time a positive inventory level and about 70% of the orders can be fulfilled directly from the current stock level.

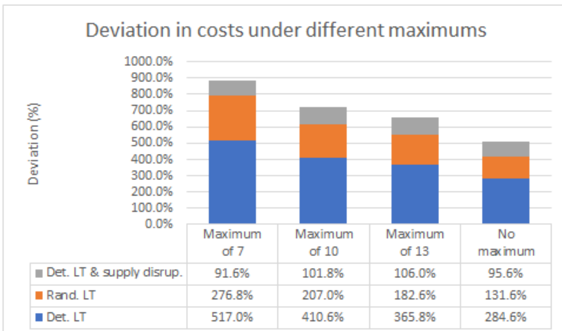

Figure 9

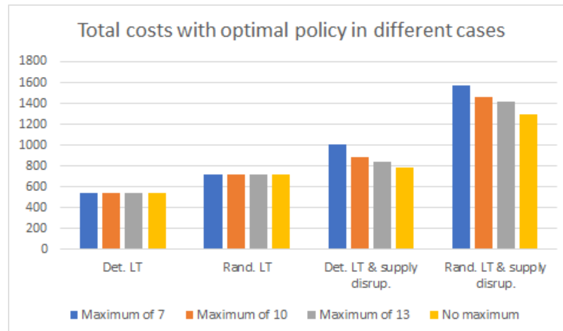

Figure 10

5 Conclusion

For an OEM or similar company it is important to know the situation they are in, for example do they need to deal with random lead times or with supply disruptions and can they receive orders while the supplier is in disruption or not. If the company is not aware of the situation the costs they need to deal with in the end can deviate a lot, especially in cases with a lot of disruption periods. For example in Figure 3 we see that if the company does not take into account random lead times they can end up with costs 89% higher. Overall we see that if the company can not receive any orders while the supplier is in disruption the effects of random lead times are higher compared to supply disruption, while if the company can receive orders it is the other way around as we can see in Figure 4. In the case of receiving orders while the supplier is in disruption, not being aware of the right situation can end up in a cost deviation of almost 300%.

To see if we can reduce the costs and the effects of random lead times and supply disruptions, we created a case with a second supplier. This supplier makes recycled parts, so their costs are less and can offer their products for less, another advantage we created for this supplier is deterministic lead times. The costs were on average 6.1% lower and the effects of the supply risks decreased as well. Last we checked what would chance is we added a maximum on the amount of a replenishment order, as expected both costs and effects increased. We even see a deviation of 517% from the combined effects, when the maximum of a replenishment order is seven.

In all the cases except using different values for $b(i)$ we see the combined effects being almost twice as much as the highest individual effect. So after all the company needs to be aware of the situation they are in to create an optimal policy and not end up with high unexpected costs.

6 Discussion

In our approach we used three different states, state 0, 1 and 2. We made the assumption that we could only changed one state per period. This assumption might be improper, if so the duration of being in state 2 would have been higher, and the chance of going into a disruption as well. The chance of going from state 0 to state 2 would have been small, because the chance of going from state 0 to state 1 already is small, so the effects would not be big but the results might have been different. And when obtaining our optimal policy we only made a distinction between the different states, but to create a more optimal policy we could have created an optimal stock level for every single time period. In our case we are dealing with a finite horizon, and the periods at the end of this horizon has a different optimal stock level than the first couple of periods. When we include the option to fulfill the orders with recycled parts we made an assumption as well, we can use 25% of the used products. But the 25% is an uncertain percentage, so the results in reality can differ from our results. When the recycling percentage is higher the total costs of using the recycling option will even decrease more, but when we can use a lower percentage of the used products the costs of using the recycling option are probably still lower than the costs without the recycling option but the difference is not significant. And for the supplier using recycled parts it will be less beneficial. We also assumed that always choosing the second-hand supplier if possible is optimal, but in state 0 the average lead time of the "normal" supplier is lower than the deterministic lead time of the second-hand supplier. So it might be possible that it is more optimal to order to the "normal" supplier in state 0 instead of ordering from the second-hand supplier.

References

V. Babich, A. N. Burnetas, and P. H. Ritchken. Competition and diversification effects in supply chains with supplier default risk. *Manufacturing & Service Operations Management*, 9(2):123–146, 2007.

J. R. Bradley and L. W. Robinson. Improved base-stock approximations for independent stochastic lead times with order crossover. *Manufacturing & Service Operations Management*, 7(4):319–329, 2005.

J. Browne. The world's most dangerous road (a bolivian detour). *Fourth Genre: Explorations in Nonfiction*, 7(2):9–19, 2005.

S. M. Disney, A. Maltz, X. Wang, and R. D. Warburton. Inventory management for stochastic lead times with order crossovers. *European Journal of Operational Research*, 248(2):473–486, 2016.

R. Ehrhardt. (s, s) policies for a dynamic inventory model with stochastic lead times. *Operations Research*, 32(1):121–132, 1984.

J. C. Hayya, T. P. Harrison, and X. J. He. The impact of stochastic lead time reduction on inventory cost under order crossover. *European Journal of Operational Research*, 211(2):274–281, 2011.

M. Hekimoğlu. *Spare Parts Management of Aging Capital Products*. Number EPS-2015-368-LIS. 2015.

M. Hekimoğlu, 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(3): 909–922, 2018.

R. S. Kaplan. A dynamic inventory model with stochastic lead times. *Management Science*, 16(7): 491–507, 1970.

X. Li, R. Dekker, C. Heij, and M. Hekimoğlu. Assessing end-of-supply risk of spare parts using the proportional hazard model. *Decision Sciences*, 47(2):373–394, 2016.

E. Lou, P. T. Mativenga, et al. What should be recycled: An integrated model for product recycling desirability. *Journal of cleaner production*, 154:51–60, 2017.

S. Özekici and M. Parlar. Inventory models with unreliable suppliersin a random environment. *Annals of Operations Research*, 91:123–136, 1999.

L. W. Robinson, J. R. Bradley, and L. J. Thomas. Consequences of order crossover under order-up-to inventory policies. *Manufacturing & Service Operations Management*, 3(3):175–188, 2001.

F. J. R. Rojo, R. Roy, and E. Shehab. Obsolescence management for long-life contracts: state of the art and future trends. *The International Journal of Advanced Manufacturing Technology*, 49 (9-12):1235–1250, 2010.

J.-S. Song and P. H. Zipkin. Inventory control with information about supply conditions. *Management science*, 42(10):1409–1419, 1996.

B. Tomlin. On the value of mitigation and contingency strategies for managing supply chain disruption risks. *Management Science*, 52(5):639–657, 2006.

P. Zipkin. Stochastic leadtimes in continuous-time inventory models. *Naval Research Logistics Quarterly*, 33(4):763–774, 1986.

Appendix

Package general: This package is the basis of the simulation, every other package links to this one. Author: Nemanja Milovanovic

- Counter.java: This class is used to model statistical counters.
- Event.java: This is an abstract class for events in a Discrete-Event Simulation.
- PerformanceMeasure.java: Abstract class representing a performance measure.
- Replication.java: This class represents one replication of a Discrete-Event Simulation
- Simulation.java: The main class of the template, allows the user to run multiple instances and compute the simulation estimates of the performance measures.
- Status.java: Class to display the current status of the simulation
- SystemState.java: Basic skeleton for a system state.
- Utils.java: This is a "static" class for library routines.

Package queue1: Package to simulate the first queue to obtain the values for P_h.

- aaDisruptionEvent.java: Makes sure what needs to happen to the system when a disruption event occurs.
- aaHealthyEvent.java: When in disruption checks if the next state is healthy, and if so changes the to healthy.

- abDepartureEvent.java: Makes sure what should happen to the system if a departure happens from the queue.
- ArrivalEvent.java: Checks every period if an arrival comes to the system, and makes the changes that are needed.
- MainQueue1.java: In this class we run the simulation and introduce the parameters, counters and performance measures.
- Queue1Replication.java: In this class the initialization happens.
- Queue1State.java: In this class we set all the parameters, variables and actions that can happen to the variables and parameters.
- TransitionProbabilities.java: In this class we calculate the transition probabilities as performance measures.
- UtilsQueue1.java: In this class we set the chances of an arrival, departure, disruption of a back to healthy event happening.

Package queue2: Package to simulate the second queue with random lead times.

- DepartureEvent.java: Makes sure what should happen to the system if the orders are being delivered.
- ArrivalEvent.java: Checks every period if an orders comes in, and changes to the system what needed.
- FillRatePM.java: Calculation for the fill rate performance measure.
- MainQueue2.java: In this class we run the simulation and introduce the parameters, counters and performance measures.
- Queue2Replication.java: In this class the initialization happens.
- Queue2State.java: In this class we set all the parameters, variables and actions that can happen to the variables and parameters.
- ReadyRatePM.java: Calculation for the ready rate performance measure.
- RecycledPM.java: Calculation for the amount of orders being fulfilled by the second-hand supplier (not used in this package).
- StateChangingEvent.java: In this class the next state (healthy or disruption) of the system is set.

- TotalCostPM.java: Calculation of the total costs as performance measures.
- UtilsQueue2.java: In this class we set the chances of an arrival, departure, state changing event happening.

Package queue2D: Package to simulate the second queue with deterministic lead times.

- DepartureEvent.java: Makes sure what should happen to the system if the orders are being delivered.
- ArrivalEvent.java: Checks every period if an orders comes in, and changes to the system what needed.
- FillRatePM.java: Calculation for the fill rate performance measure.
- MainQueue2.java: In this class we run the simulation and introduce the parameters, counters and performance measures.
- Queue2Replication.java: In this class the initialization happens.
- Queue2State.java: In this class we set all the parameters, variables and actions that can happen to the variables and parameters.
- ReadyRatePM.java: Calculation for the ready rate performance measure.
- StateChangingEvent.java: In this class the next state (healthy or disruption) of the system is set.
- TotalCostPM.java: Calculation of the total costs as performance measures.
- UtilsQueue2.java: In this class we set the chances of an arrival, departure, state changing event happening.

Package queue2New: Package to simulate the second queue with random lead times, and including transportation failure.

- DepartureEvent.java: Makes sure what should happen to the system if the orders are being delivered, including the chance of transportation failures.
- ArrivalEvent.java: Checks every period if an orders comes in, and changes to the system what needed.
- FillRatePM.java: Calculation for the fill rate performance measure.

- MainQueue2.java: In this class we run the simulation and introduce the parameters, counters and performance measures.
- Queue2Replication.java: In this class the initialization happens.
- Queue2State.java: In this class we set all the parameters, variables and actions that can happen to the variables and parameters.
- ReadyRatePM.java: Calculation for the ready rate performance measure.
- StateChangingEvent.java: In this class the next state (healthy or disruption) of the system is set.
- TotalCostPM.java: Calculation of the total costs as performance measures.
- UtilsQueue2.java: In this class we set the chances of an arrival, departure, state changing event happening.

Package queue2R: Package to simulate the second queue with random lead times, and the option of a second-hand spare parts supplier.

- DepartureEvent.java: Makes sure what should happen to the system if the orders are being delivered.
- ArrivalEvent.java: Checks every period if an orders comes in, and changes to the system what needed.
- FillRatePM.java: Calculation for the fill rate performance measure.
- MainQueue2.java: In this class we run the simulation and introduce the parameters, counters and performance measures.
- Queue2Replication.java: In this class the initialization happens.
- Queue2State.java: In this class we set all the parameters, variables and actions that can happen to the variables and parameters.
- ReadyRatePM.java: Calculation for the ready rate performance measure.
- RecycledPM.java: Calculation for the amount of orders being fulfilled by the second-hand supplier (not used in this package).
- RecycleEvent.java: In this class the orders is delivered by the second-hand supplier, and the changes need to the system are made.

- StateChangingEvent.java: In this class the next state (healthy or disruption) of the system is set.
- TotalCostPM.java: Calculation of the total costs as performance measures.
- UtilsQueue2.java: In this class we set the chances of an arrival, departure, state changing event happening.

Package queue2RD: Package to simulate the second queue with deterministic lead times, and the option of a second-hand spare parts supplier.

- DepartureEvent.java: Makes sure what should happen to the system if the orders are being delivered.
- ArrivalEvent.java: Checks every period if an orders comes in, and changes to the system what needed.
- FillRatePM.java: Calculation for the fill rate performance measure.
- MainQueue2.java: In this class we run the simulation and introduce the parameters, counters and performance measures.
- Queue2Replication.java: In this class the initialization happens.
- Queue2State.java: In this class we set all the parameters, variables and actions that can happen to the variables and parameters.
- ReadyRatePM.java: Calculation for the ready rate performance measure.
- RecycledPM.java: Calculation for the amount of orders being fulfilled by the second-hand supplier (not used in this package).
- RecycleEvent.java: In this class the orders is delivered by the second-hand supplier, and the changes need to the system are made.
- StateChangingEvent.java: In this class the next state (healthy or disruption) of the system is set.
- TotalCostPM.java: Calculation of the total costs as performance measures.
- UtilsQueue2.java: In this class we set the chances of an arrival, departure, state changing event happening.

AlgorithmInventoryCoverage.m: In this program we calculate the chance of an order being delivered in a certain amount of time periods.