BSc Thesis: Scheduling policies for a repair shop problem

Dopheide, J.J.
Student Number: 371175
July 3, 2016

Econometrics and operations research
Supervisor: Oosterom, C.D. van
First Reader: Dekker, R.

Abstract

In this paper we take another look at the Myopic(**R**) scheduling policy proposed by Liang et al. (2013). The Myopic(**R**) policy is much more time efficient, coming with a small loss of optimality. This policy is useful when one repair shop for every machine type is combined to one central repair shop. The other case Liang et al. (2013) look at is the base case, where one repair shop per fleet is used.

In this paper multiple search methods are proposed to find the optimal costs for both the base case as the central repair shop case under the Myopic(**R**) policy. These search methods are based on the convexity of the cost functions of the base case and the CRS case. which is proven for the first and conjectured for the second. The proposed methods show great improvement in calculation time.

In Liang et al. (2013) it is shown that the CRS is preferred over the base case for certain a certain factor α. We show for which values the cost minima of the base case and the CRS case are in equilibrium., which means that we know when it is benificial to use which repair shop strategy.

Contents

1 Introduction

In a production line, the manufacturing of products is often partitioned into several stages where in every stage a different kind of machine is used. Within each stage the same type of machine is used. All the machines together in one stage are then called a fleet. Machines break down from time to time and have to be repaired. When a machine is down the operation targets cannot be met, so until the machine is repaired, the manufacturing plant experiences downtime-costs. To decrease the downtime-costs, it can be useful to have spare machines in stock, so the broken ones can be replaced immediately, and the production process is not interrupted. These spares are accompanied by holding costs per type of machine. Therefore, it is of interest to determine a good balance between the number of spares and the long-term expected downtime, so the costs for production are at its minimum. Also, the repair shop structure is of importance. A manufacturer could use a different repair shop per fleet, or a general repair shop for all the different fleets together (with a higher service (repair) rate). Sahba et al. (2013) shows that a central repair shop can be more efficient.

Sahba et al. (2013) looks at the option where there are spares kept per fleet, but also the case where there is a general stock where all fleets can draw resources from. This last case is beyond the scope of this research. When using a central repair shop, the scheduling of the repairs is of importance as well. This is caused by the fact that the downtime costs for every machine is different. The machine that the decreases the downtime costs the most when repaired relatively to the repair time, has to be repaired first to save as much as possible.

In this paper the Myopic(**R**) policy proposed by Liang et al. (2013) will be re-examined. The Myopic(**R**) policy is very useful, since, according to the results in Liang et al. (2013), the computational time decreases drastically compared to time required to find the optimal solution, while outcomes do not differ a lot from the optimal solution. The minimum costs are determined by searching over different number of spares in stock per machine type, and calculating the system costs for those numbers. The costs of the central repair shop while using the Myopic(**R**) will be compared to the one repair shop per fleet case.

Since the Myopic(**R**) policy is introduced to decrease computational time, it is of interest to use a clever way of finding the optimal number of spares as well, to decrease the computational time even more. In Liang et al. (2013) the way to find the optimal number of spares is undefined and therefore we assume that they used complete enumeration, which is slow. Very few literature could be found regarding keeping spare inventory for breaking down machines. This is said in Liang et al. (2013) as well. Therefore the purpose of this research is to make sure we find the optimal value of stock and find a method to decrease the computational time of finding the optimal number of spares kept in stock per fleet.

Intuitively, the system cost function for the number of spares is first non-increasing and then increasing. This means that there is a global minimum for the system costs. This intuitive assumption is supported by the fact that adding spares has a constant increase in costs, since the holding costs per machine type are constant. Furthermore, increasing the number of spares, causes a decrease in probability of having less machines operational than required, and therefore the downtime costs are non-increasing. Taylor and Jackson (1954) results show that probability of machines is the number of spare machines in the one repair shop per fleet case is strictly decreasing descending in their graphs, which may be the case in the central repair shop case as well. This assumption has to be checked and may be the basis for a clever search algorithm to find the optimal number of spares per fleet. We will prove that for the multiple repair shop case the cost function is indeed convex and use this result to propose two fast ways of finding the optimal number of spares per fleet. We will show by graphs that the (marginal) cost functions in the central repair shop case are also convex for the cases we examine and will propose search methods for finding the optimal number of spares.

Also, Liang et al. (2013) show that a central repair shop is more cost efficient than separate repair shops under the assumption of higher repair rates per fleet. In their paper they assumed when two fleets are repaired in the same repair shop, their repair times increase with factor two. We assume that the central repair shop case is indeed beneficial over the base case when all the repair rates are multiplied by factor r when r repair shops are combined in one repair shop. Another purpose of this research is to find out for what factor α the central repair shop is not preferred over the base case any more. In other words, we want to find out for which factor α the optimal system costs for the central repair shop and the separate shops are in equilibrium.

It is of interest to find this equilibrium to know for which production lines a central repair shop should be used, and for which lines a separate repair shop is beneficial. Also a hybrid method could be used, such that some machine types are repaired in the same repair shop and others have its own repair shop or are combined in another separate repair shop.

2 Notation

To reproduce the case where we use a different repair shop per fleet of machines and the central repair shop case while using the Myopic(**R**) scheduling policy, we first introduce notation. We use similar notation asLiang et al. (2013). From now on we will refer to the repair shop per fleet as the base case (BC), and to the combined repair shop as the central repair shop (CRS).

We consider r different fleets with identical machines types i, $i = 1, 2, .., r$ within each fleet. The number of required machines within a fleet to operate at full strength is denoted as N_i. Since we allow to keep spares per fleet as well, the number of spares kept in stock per fleet is defined as S_i. The total number of machines of type i owned by the manufacturer equals $N_i + S_i$.

The number of machines that is operational and active at time t is defined as $W_i(t)$, where $0 \leq W_i(t) \leq N_i$. The number of functional machines that are kept in stock at time t is defined as $I_i(t)$, where $0 \leq I_i(t) \leq S_i$. Since we will always operate at the highest operation level, such that all functional will be used if possible. This means that $I_i(t) = 0$ when $W_i(t) < N_i$. The total number of functional machines is defined as $A_i(t) = W_i(t) + I_i(t)$. When A_i is equal or larger than N_i, fleet i operates at full strength. When A_i is smaller than N_i the fleet has $N_i - A_i$ down machines.

When there are less machines operational than required to operate at full strength ($W_i(t) < N_i$), the manufacturer experiences less production than is optimal. This results in less turnover/profit, which means that the manufacturer experiences extra costs. Therefore we define b_i as the costs per time unit per machine that is down. The total holding costs per time unit is $h_i \times S_i$, where h_i are the holding costs per time unit per machine. The downtime costs and holding costs are assumed to be constant per machine type.

We say that a system i is in state n_i, when there are n_i machines functional. A machine from fleet i is repaired with a rate μ_i and a machine breaks down with rate λ_i. The break down rate is state dependent, since the expected time till the first break down is dependent on the number of machines that are in use. The break down rate in state n_i equals $\lambda_i \cdot \min\{n_i, N_i\}$.

3 Methodology

3.1 Base Case

We will first look into the BC. The BC can be modeled as a birth-and-death process. A birth takes place when a machine is repaired and a death takes place when a machine breaks down. Therefore a birth takes place with rate μ_i and a death takes place with λ_i.

In the BC every fleet has its own repair shop. This means that the probability of having n functional machines in fleet i, is independent of the number of machines functional in fleet j, where $i \neq j$. Therefore, we can consider this model as r different queuing systems. Every repair shop has its own state space $\mathcal{S}$ that consists of the numbers $1, 2, ..., N_i + S_i$, corresponding to the number of functional machines in every state. Since the fleets are independent in the BC we will solve the different systems one by one, and from now on we drop the subscript i for notational convenience.

Now we can define the probability that the system is in state n and we define the probability of being in state n while having S spares in stock as follows. Because the state space is dependent on the number of spares S_i, the state probabilities are dependent on S_i as well. Therefore, we use the following notation:

$$p(n|S) = \frac{\frac{\mu^n}{\lambda^n \cdot \prod_{i=1}^{n} \min\{i,N\}}}{\sum_{n=0}^{N+S} \frac{\mu^n}{\lambda^n \cdot \prod_{i=1}^{n} \min\{i,N\}}} \tag{1}$$

We need to construct the cost function, because we want to minimize the total production costs, such that:

$$C^*_{BC} = \sum_{i=1}^{r} C_i(S^*_i)$$

is minimal.

The state probabilities in equation 1 are an important part here. The cost function is defined as follows:

$$C(S) = h \cdot S + b \sum_{n=0}^{N} (N - n) p(n|S) \tag{2}$$

The first term are the total holding costs for the machines kept in stock, and the second term are the costs for the down machines. We need to minimize this function for every fleet i to determine the minimum costs.

To find the optimal value of S, for which the function $C(S)$ is at its minimum, we compute the costs for different values of S. In Liang et al. (2013) the search method is undefined and therefore we assumed that they used complete enumeration with a predefined maximum value of S. The choice of this maximum S is also undefined. The choice of a maximum S^{max} is hard, because taking S^{max} large, means that a lot of different options have to be considered and computed, when complete enumeration is used. This takes a lot of computation time. Taking S^{max} small, might cause that the global minimum of the cost function is not included in the interval, such that $S^* \notin \{0, 1, ..., S^{max}\}$. Therefore the optimal S cannot be found when S^{max} is too small. This means that it is of great interest to find out if there are other possibilities to find S^*.

3.1.1 Convexity

As mentioned in Section 1, we want to show that the cost function is convex in the BC. Convexity in general for an integer function $y : \mathbb{Z} \to \mathbb{R}, \forall x \in \mathbb{Z}$ is defined as follows:

$$y(x+1) - y(x) \leq y(x+2) - y(x+1)$$

We will prove that the function given in 2 is convex.

Theorem 1: The cost function of the base case is convex, or equivalently:

$$C(S+1) - C(S) \leq C(S+2) - C(S+1) \tag{3}$$

To prove this theorem we will first prove the following lemma.

Lemma 1: The probability function of the system being in state n given a stock S is convex in S:

$$p(n|S+1) - p(n|S) \leq p(n|S+2) - p(n|S+1) \quad \forall n \tag{4}$$

Proof of Lemma 1: For notational convenience we will first introduce $\phi_n = \frac{\mu^n}{\lambda^n \cdot \left(\frac{n!}{\prod_{i=N+1}^{n} i} \cdot \prod_{i=1}^{n-N} (N) \right)}$ such that Equation 1 becomes $p(n|S) = \frac{\phi_n}{\sum_{n=0}^{N+S} \phi_n}$. First we will work out the left hand side of the equation and next the right hand side.

LHS:

$$\begin{aligned} p(n|S+1) - p(n|S) &= \frac{\phi_n}{\sum_{n=0}^{N+S+1} \phi_n} - \frac{\phi_n}{\sum_{n=0}^{N+S} \phi_n} \\ &= \frac{\phi_n}{\sum_{n=0}^{N+S+1} \phi_n} - \frac{\frac{\sum_{n=0}^{N+S+1} \phi_n}{\sum_{n=0}^{N+S} \phi_n}}{\frac{\sum_{n=0}^{N+S+1} \phi_n}{\sum_{n=0}^{N+S} \phi_n}} \cdot \frac{\phi_n}{\sum_{n=0}^{N+S} \phi_n} \\ &= \frac{\phi_n}{\sum_{n=0}^{N+S+1} \phi_n} - \frac{\frac{\sum_{n=0}^{N+S+1} \phi_n}{\sum_{n=0}^{N+S} \phi_n} \cdot \phi_n}{\sum_{n=0}^{N+S+1} \phi_n} \\ &= \frac{\left(1 - \frac{\sum_{n=0}^{N+S+1} \phi_n}{\sum_{n=0}^{N+S} \phi_n}\right) \cdot \phi_n}{\sum_{n=0}^{N+S+1} \phi_n} \\ &= \frac{\left(\frac{\sum_{n=0}^{N+S} \phi_n}{\sum_{n=0}^{N+S} \phi_n} - \frac{\sum_{n=0}^{N+S+1} \phi_n}{\sum_{n=0}^{N+S} \phi_n}\right) \cdot \phi_n}{\sum_{n=0}^{N+S+1} \phi_n} = \frac{\left(-\frac{\phi_{N+S+1}}{\sum_{n=0}^{N+S} \phi_n}\right) \cdot \phi_n}{\sum_{n=0}^{N+S+1} \phi_n} \end{aligned}$$

RHS:

$$p(n|S+2) - p(n|S+1) = \cdots = \frac{\left(-\frac{\phi_{N+S+2}}{\sum_{n=0}^{N+S+1} \phi_n}\right) \cdot \phi_n}{\sum_{n=0}^{N+S+2} \phi_n} = \frac{\left(-\frac{\phi_{N+S+2}}{\sum_{n=0}^{N+S+2} \phi_n}\right) \cdot \phi_n}{\sum_{n=0}^{N+S+1} \phi_n}$$

We now have to show that:

$$\frac{\left(-\frac{\phi_{N+S+1}}{\sum_{n=0}^{N+S}\phi_n}\right)\cdot\phi_n}{\sum_{n=0}^{N+S+1}\phi_n} \leq \frac{\left(-\frac{\phi_{N+S+2}}{\sum_{n=0}^{N+S+2}\phi_n}\right)\cdot\phi_n}{\sum_{n=0}^{N+S+1}\phi_n}$$

Or equivalently:

$$-\frac{\phi_{N+S+1}}{\sum_{n=0}^{N+S}\phi_n} \leq -\frac{\phi_{N+S+2}}{\sum_{n=0}^{N+S+2}\phi_n} \tag{5}$$

To show this inequality we need to substitute ϕ_n back into the equation:

$$\begin{aligned}
-\frac{\phi_{N+S+1}}{\sum_{n=0}^{N+S}\phi_n} &= -\frac{\frac{\mu^{N+S+1}}{\lambda^{N+S+1}\cdot\prod_{i=1}^{N+S+1}\min\{i,N\}}}{\sum_{n=0}^{N+S}\frac{\mu^n}{\lambda^n\cdot\prod_{i=1}^{n}\min\{i,N\}}} \\
&= -\frac{\left(\frac{\mu}{\lambda}\right)^{N+S+1}}{\sum_{n=0}^{N+S}\left(\frac{\mu}{\lambda}\right)^n\cdot\left[\frac{\prod_{i=1}^{N+S+1}\min\{i,N\}}{\prod_{i=1}^{n}\min\{i,N\}}\right]} && (6)\\
&\leq -\frac{\left(\frac{\mu}{\lambda}\right)^{N+S+1}}{\sum_{n=0}^{N+S}\left(\frac{\mu}{\lambda}\right)^n\cdot\left[\frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n+1}\min\{i,N\}}\right]} && (7)\\
&\leq -\frac{\left(\frac{\mu}{\lambda}\right)^{N+S+1}}{\frac{\lambda}{\mu}\prod_{i=1}^{N+S+2}\min\{i,N\}+\sum_{n=0}^{N+S}\left(\frac{\mu}{\lambda}\right)^n\cdot\left[\frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n+1}\min\{i,N\}}\right]} && (8)\\
&= -\frac{\left(\frac{\mu}{\lambda}\right)\cdot\left(\frac{\mu}{\lambda}\right)^{N+S+1}}{\prod_{i=1}^{N+S+2}\min\{i,N\}+\sum_{n=0}^{N+S}\left(\frac{\mu}{\lambda}\right)^{n+1}\cdot\left[\frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n+1}\min\{i,N\}}\right]} \\
&= -\frac{\left(\frac{\mu}{\lambda}\right)^{N+S+2}}{\sum_{n=0}^{N+S+1}\left(\frac{\mu}{\lambda}\right)^n\cdot\left[\frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n}\min\{i,N\}}\right]} \\
&= -\frac{\frac{\mu^{N+S+2}}{\lambda^{N+S+2}\cdot\prod_{i=1}^{N+S+2}\min\{i,N\}}}{\sum_{n=0}^{N+S+1}\frac{\mu^n}{\lambda^n\cdot\prod_{i=1}^{n}\min\{i,N\}}} = -\frac{\phi_{N+S+2}}{\sum_{n=0}^{N+S+2}\phi_n}
\end{aligned}$$

The inequality between 6 and 7 is explained by the fact that:

$$\begin{aligned}
\frac{\prod_{i=1}^{N+S+1}\min\{i,N\}}{\prod_{i=1}^{n}\min\{i,N\}} &= \frac{\frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n+1}\min\{i,N\}}}{\frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n+1}\min\{i,N\}}}\cdot\frac{\prod_{i=1}^{N+S+1}\min\{i,N\}}{\prod_{i=1}^{n}\min\{i,N\}} \\
&= \frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n+1}\min\{i,N\}}\cdot\frac{\min\{n+1,N\}}{\min\{N+S+2,N\}} \\
&\leq \frac{\prod_{i=1}^{N+S+2}\min\{i,N\}}{\prod_{i=1}^{n+1}\min\{i,N\}}
\end{aligned}$$

The inequality between 7 and 8 is explained by the fact that the term $\frac{\lambda}{\mu}\prod_{i=1}^{N+S+2}\min\{i,N\}$ is always positive, since $\lambda,\mu,N > 0$. Therefore inequality 5 holds and **Lemma 1** (4) is proven. □

Lemma 1 will be useful for proving **Theorem 1**.

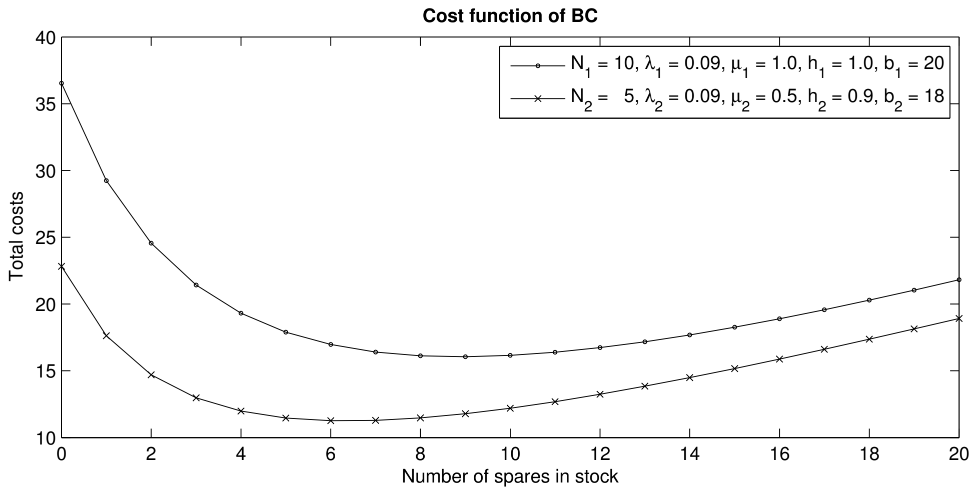
Proof of Theorem 1:

$$
\begin{aligned}
C(S+1) - C(S) &= \left(h(S+1) + b\sum_{n=0}^{N}(N-n)p(n|S+1)\right) - \left(hS + b\sum_{n=0}^{N}(N-n)p(n|S)\right) \\
&= h + b\sum_{n=0}^{N}(N-n)\Big[p(n|S+1) - p(n|S)\Big] \qquad (9) \\
&\leq h + b\sum_{n=0}^{N}(N-n)\Big[p(n|S+2) - p(n|S+1)\Big] \qquad (10) \\
&= \left(h(S+2) + b\sum_{n=0}^{N}(N-n)p(n|S+2)\right) - \left(h(S+1) + b\sum_{n=0}^{N}(N-n)p(n|S+1)\right) \\
&= C(S+2) - C(S+1)
\end{aligned}
$$

The inequality between 9 and 10 holds, because of **Lemma 1**. This proofs **Theorem 1** (3) that the cost function of the BC is convex. □

The convexity of the BC cost function will be very useful in proposing a method that reduces the computation time for finding the optimal value of S, and still guarantees that we find S^*, the number of spares in stock for which the operation costs are at its global minimum.

Figure 1: Cost functions for two fleets with its own repair shops (BC)



In Figure 1 an example of the cost function is shown. The convexity is very clear for these two functions.

3.1.2 Search methods

The first method we propose to find S^* uses the property of a convex function, that when it starts increasing, it will never decrease anymore. The algorithm computes the costs for every S, until the cost function value starts increasing. This means that the previous iteration corresponds to the number of spares S at which the production costs are at its minimum. The pseudo-code for this algorithm is as follows:

Pseudo Code Base Case 1

Set $C(0) \rightarrow \inf$
Set $S \rightarrow 0$
repeat
  $k \rightarrow S + 1$
  Compute $C(S)$
until $C(S) > C(S-1)$;
$C^* \rightarrow C(S-1)$
$S^* \rightarrow S - 1$
Return S^*, C^*

In Figure 1 we show an example of the cost functions for two fleets (corresponding to the first row of results in Appendix 7). If we use complete enumeration, and we set S_{max} for both fleets at 20, 40 function evaluations have to be computed. Since the minima are at $S_1 = 9$ and $S_2 = 6$, the algorithm we propose only uses 19 function evaluations. Another advantage is that it costs more time to solve a system of equations when the number of equations and variables increase. When S is large, there are more states in which the system can be and therefore there are more variables and equations. Therefore it is beneficial to compute the cost function only for small values of S, which is done in the proposed method.

Another search method that is proposed is the Fibonacci search method. The Fibonacci is similar to the well-known golden search algorithm, but is suitable for an integer function such as the cost function 2 in Section 3.1. The Fibonacci search algorithm is a four point search method that shrinks a start interval iteratively wherein the minimum of the cost function will be. In the end the interval consists of four function values and the minimum will be searched over these four values, and S^* will be the corresponding stock to this minimum.

The advantage of this method is that function evaluations get reused, just as in the golden search method, which saves computation time. The downside of this search method is that it needs an upper bound. We did not look into how to find a upper bound that guarantees that the minimum will indeed lie within the interval.

The pseudo-code of the algorithm is shown on the next page. We define F_i as the i^{th} Fibonacci number and F_n as the smallest Fibonacci number greater than the preset upper bound. The new upper bound of the interval will be F_n, such that the minimum is searched in the interval $[0, F_n]$.

Pseudo Code Base Case 2

```
Set L → 0
Set U → F_n
Set C_L, C_U → inf
Set S_1 → F_{n−2}
Set S_2 → F_{n−1}
Compute C_1 → C(S_1)
Compute C_2 → C(S_2)
for j = 3, ..., n − 1 do
    if C_1 < C_2 then
        Set U → S_2
        Set C_U → C_2
        Set S_2 → S_1
        Set C_2 → C_1
        Set S_1 → L + F_{n−j}
        Compute C_1 → C(S_1)
    else
        Set L → S_1
        Set C_L → C_1
        Set S_1 → S_2
        Set C_1 → C_2
        Set S_2 = U − F_{n−j}
        Compute C_2 = C(S_2)
    end
end
if C_L equals inf then
    Compute C_L → C(L)
end
if C_L equals inf then
    Compute C_U → C(U)
end
Set C* → min({C_L, C_1, C_2, C_U})
Set S* to corresponding S
Return C*, S*
```

3.2 Central Repair Shop

In the Central Repair Shop case (CRS), all machines are repaired at the same shop, but with a higher repair rate. For now we assume that $\mu_i = r \cdot \mu_i^{BC}$. Since every machine is repaired at the same shop, the state probabilities for every fleet are not independent anymore. This means that the problem becomes much larger and costs more time to solve.

Let us first define the vector $\mathbf{n} = (n_1, n_2, ..., n_r)$, which states the number of functional machines of every fleet. We also define vector $\mathbf{S} = (S_1, S_2, ..., S_r)$, which states the number of spares kept in stock per fleet. Our goal is to find the optimal vector $\mathbf{S}$ for which the operation costs for the CRS are at its minimum. Therefore we need to minimize the cost function, which is slightly different in the CRS compared to the BC, due to the dependence between the fleets:

$$C^* = \min_{\mathbf{S}} \left\{ \sum_{i=1}^{r} C_i(\mathbf{S}) \right\}$$

where,

$$C_i(\mathbf{S}) = h_i \cdot S_i + b_i \sum_{n=0}^{N_i} (N_i - n) p_i(n)$$

The minimum costs depend on the scheduling policy that is used in the CRS. The scheduling policy states which machine type is going to be repaired when not all machines are functional. Liang et al. (2013) propose the Myopic(**R**) policy, which is faster than the optimal scheduling, and has a low cost difference with the optimal scheduling policy.

3.2.1 Myopic(R) policy

The Myopic(**R**) policy chooses to repair the machine type first for which the cost difference relatively to its repair and breakdown rate, between repairing and not repairing this machine is the largest. This cost rate difference is dependent on the number of type i machines in stock in a certain state. Therefore we define x_i as the inventory position of machine type i. The cost difference rate is defined as follows for $x_i > 0$:

$$\Delta c_i^R(x_i) = -b_i \sum_{m=x_i+1}^{N_i+x_i} p_i^R(m) \tag{11}$$

The probability $p_i^R(m)$ is the probability that m machines will break during the look ahead time. The look ahead time in the Myopic(**R**) policy is the repair time. Liang et al. (2013) show that for $0 \leq x_i \leq m$:

$$p_i^R(m) = \frac{\frac{N_i!}{(N_i-m+x_i)!} \frac{\mu_i}{\lambda_i} \frac{(N_i\lambda_i)^{x_i}}{(N_i\lambda_i+\mu_i)^{x_i}}}{\prod_{j=0}^{m-x_i} (N_i + \frac{\mu_i}{\lambda_i} - j)} \tag{12}$$

When $x_i = 0$, there can be precisely N_i functional machines, or less. When $x_i = 0$ and there are exactly $n_i = N_i$ machines functional , the costs are defined as:›

$$c_i^R(x_i) = -b_i \cdot \frac{(N_i\lambda_i)}{N_i\lambda_i + \mu_i} \tag{13}$$

When $x_i = 0$ and the system is not fully operational, such that $n_i < N_i$, the expected downtime rate costs are defined as:

$$c_i^R(x_i) = -b_i \tag{14}$$

For every state **n** we can determine which machine has the lowest expected downtime cost rate difference. The machine type that corresponds to this minimum for $\frac{\mu_i \cdot \Delta c_i^R(x_i)}{\lambda_i}$ is the machine that will be repaired in this state. As can be seen, we compensate for the repair and break down rate in this fraction, such that we look at the relative downtime costs. Since we now know which machine will be repaired in every state we can compute the global balance equations:

3.2.2 Global balance equations

To find the long-term state probabilities, we compose a system of equations.

The left hand side of the equation (first line) is the rate that the process leaves state $\mathbf{n} = (n_1, n_2, ..., n_r)$.

The first term of the right hand side (second line) is the rate that the process enters state $\mathbf{n} = (n_1, n_2, ..., n_r)$ by a break down of any of the machines.

The second term of the right hand side (third line) is the rate that the process enters state $\mathbf{n} = (n_1, n_2, ..., n_r)$ by a repair.

$$\left[\sum_{i=1}^{r} \lambda_i(n_i) + \mu(\mathbf{n})\right] P(\mathbf{n}) =$$
$$\sum_{i=1}^{r} [\lambda_i(n_i + 1) \cdot P(\mathbf{n} + \mathbf{e}_i)] +$$
$$\sum_{i=1}^{r} [\mu(\mathbf{n} - \mathbf{e}_i) \cdot I_i \cdot P(\mathbf{n} - \mathbf{e}_i)]$$

where $\mathbf{e}_i$ is a unit-vector.

Where

$$\lambda_i(n_i) = \min\{n_i, N_i\} \cdot \lambda_i$$

is the failure rate of a type i machine given that there are $\min\{n_i, N_i\}$ functional and active machines.

Where

$$\mu(\mathbf{n}) = \begin{cases} \mu_i, & \text{if not all machines are functional} \\ 0, & \text{otherwise} \end{cases}$$

where i is the to be repaired machine based on the Myopic(**R**) scheduling policy in Section 3.2.1.

Where

$$I_i = \begin{cases} 1, & \text{if type } i \text{ machine in function } \mu(.) \text{ is scheduled for repair} \\ 0, & \text{otherwise} \end{cases}$$

At last we add the normalizing equation, such that the total probability equals one:

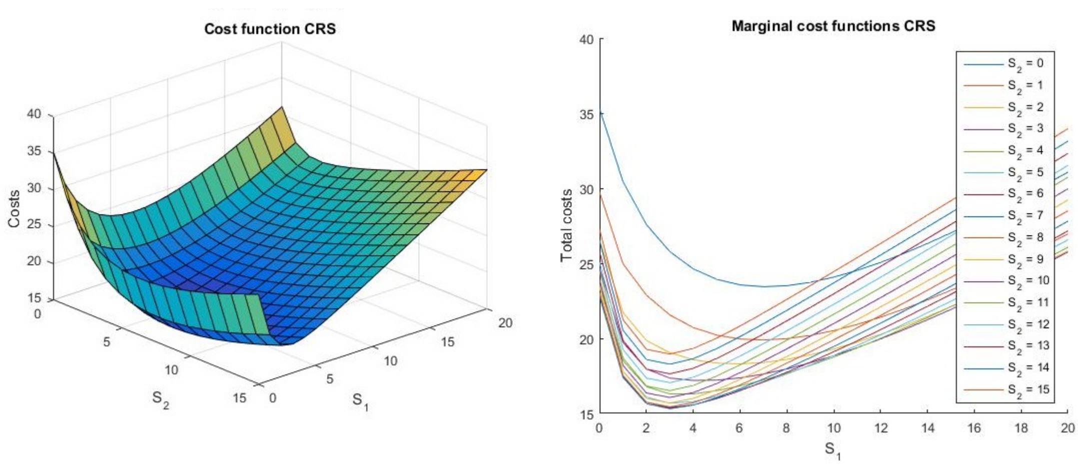
$$\sum_{\mathbf{n}} P(\mathbf{n}) = 1$$

We will solve this system by using the MATLAB function for solving a system of equations.

3.2.3 Search methods

Intuitively the CRS cost function is convex, just as the BC cost function. When the multiple stocks S_i increase, the downtime costs are expected to be decreasing descending, and therefore expected to be convex in $\mathbf{S}$. The holding costs increase linear, so are convex in $\mathbf{S}$ as well. Since the sum of two convex function is also convex, the the CRS cost function has to be convex as well if the assumption on the downtime costs holds. We could not prove this assumption for the CRS, since it is too complex for the time span we have.

Figure 2: Cost functions for two fleets sharing the same repair shop (CRS)



Therefore we checked this assumption by numerical experiments. In Figure 2 the cost function is shown for the experiment corresponding to the first row of the results in Appendix 7. As we expected we can see in both graphs clear convex behavior. Although we do not know for sure if convexity holds in every case, we propose an algorithm which will decrease the computational time drastically if this assumption indeed holds.

The algorithm is the same as the first proposed method in the BC for a given stock for all the fleets except for fleet 1, so for a given $S_2, ..., S_r$. This means the *stop at first increase*-method is used in one dimension. In the other dimensions we use complete enumeration to guarantee convergence to the Myopic($\mathbf{R}$) optimum.

If the cost function of the Myopic($\mathbf{R}$) is convex, this method will converge. Since we have no proof of the convexity of this function, there is no guarantee that the optimal vector $\mathbf{S}$ will be found. Therefore this algorithm is a heuristic for the Myopic($\mathbf{R}$) policy in the CRS.

Another method we propose, is also based on the assumption of convexity in the cost function. It is almost the same as the previous, but were we first only used the *stop at first increase*-method in one direction, we will now use it in all directions. This method is also known as the *coordinate descent*-method.

The algorithm will search for an optimal value for S_i given S_j, $\forall j \neq i \in R$, where $R = \{1, 2, ..., r\}$. We will use the line search method from Section 3.1.2 to find S_i^*. When the optimal value for S_i is found, another direction i will be chosen, and another optimum will be searched for along the new line. The process will repeat itself until no more improvement is found. The algorithm will search in the following order of directions: $i = 1, 2, ..., r, 1, 2, ..., r....$

A slight adjustment has to be notified according to the line search method, because we used to start at the search at $S_i^0 = 0$, such that we only have to increase S_i to find the minimum costs. Now that $S_i^0 \geq 0$ in each line search, the optimum can be on either sides of the current S_i. We propose to first look if a decrease is found when S_i is increased, and if not so, we search in the other direction along the line.

The downside of this method that it does not guarantee convergence to the minimum, even for convex functions. The upside is that this method will be faster since the information from the previous line-search will be used in the next.

3.3 BC/CRS-equilibrium

In the BC there is one repair shop per machine type, and in the CRS there is only one repair shop. This means that the CRS has more repairs to handle than the BC, since the break down rate per shop in the BC is $n_i \cdot \lambda_i, \forall i$, and in the CRS this rate equals $\sum_{i=1}^{r} n_i \cdot \lambda_i$. The CRS can only handle one machine at the time and the BC at most r different machines at the same time.

For trivial reasons, the BC will therefore be lower in operational costs when $\mu_i^{CRS} = \mu_i^{BC}$. Keeping the repair rates at the same in the BC and the CRS, does not make sense, since you will just lose workforce. In Liang et al. (2013), they use $\mu_i = 2 \cdot \mu_i^{BC}$, when $r = 2$. Therefore we assume that they used $\alpha = r$ in $\mu_i^{CRS} = \alpha \cdot \mu_i^{BC}$. This makes sense when for example in every shop there is an equal number of repairmen, and the types of machines do not require a specific skill set. The total workforce is then combined in the central repair shop resulting in $\alpha = r$.

There are also a lot of reasons to argue the $\alpha = r$ assumption. For example, if a machine type does need a specific skill to be repaired, combining workforce from other repair shops in one central repair shop will result not in $\mu_i = r \cdot \mu_i^{BC}$, since the workforce from a repair shop j, where $j \neq i$, cannot repair machine i at the same rate as the workforce from repair shop i.

Because the $\alpha = r$ assumption may be argued, we are going to look for which value of α the operation costs in the CRS are equal to the costs in the BC. We will do this to search over different values of α using the golden search method. For $r = 2$, we use the initial interval of $\alpha = [1, 2]$, because at $\alpha = 1$ the BC is preferred over the CRS and as shown by Liang et al. (2013) the CRS is preferred when $\alpha = 2$, with $r = 2$. Furthermore the function we evaluate is the absolute value of the difference between the CRS and BC operation costs ($f(\alpha) = |C_{BC} - C_{CRS}(\alpha)|$, where $C_{CRS}(\alpha)$ is the cost function of the CRS, with $\mu_i = \alpha \cdot \mu_i^{BC}$). Since intuitively we know the minimum of this function equals 0, the stopping criterium we use is $f(\alpha) < 10^4$. We also set the maximum number of iterations on 25.

4 Results

Before we take a look at the results of the several , we have to introduce numerical examples. We will use the same instances as Liang et al. (2013) use. We consider a production line with $r = 2$ fleets, with the following parameters:

- Setting $h_1 = 1$, we consider the following holding cost rates for fleet 2: $h_2 = \{0.9, 0.7, 0.5\}$;
- We consider the following down time cost rate to holding cost rations: $\frac{b_1}{h_1} = \frac{b_2}{h_2} = \{20, 30\}$;
- The fleet sizes are: $(N_1, N_2) \in \{(10, 5), (10, 10, (10, 15), (50, 25), (50, 50), (100, 50)\}$;

- When $N_1 = 10, 100$ ($N_1 = 50$), we set $\mu_1 = 2$ ($\mu_1 = 1$), and we consider $\frac{\mu_1}{\mu_2} \in \{2, 1, \frac{2}{3}\}$;
- As an approximate measure of the repair shop utilization, we set $u = \frac{\lambda_1 N_1}{\mu_1} = \frac{\lambda_2 N_2}{\mu_2} \in \{0.45, 0.35, 0.25\}$ corresponding to high, medium and low levels of repair shop utilization.

This results in a total of $3 \times 2 \times 6 \times 3 \times 3 = 324$ instances which have to be computed for all methods. The results can be found in Appendix 7.

The results of these numerical experiments did agree with the results in Liang et al. (2013), except for the *coordinate descent*-method. This is caused by the fact that it is a heuristic that does not always converge to the Myopic(**R**) optimum. For that reason the *coordinate descent*-method has its own cost and optimal stock columns in the tables with results.

4.1 Base Case

When we use complete enumeration we have to set an S^{max} to find the optimal number of spare machines. For the BC we used an $S^{max} = 50$ for both fleets, which we assumed to be an arbitrarily large enough number.

In Section 3.1 we propose two search methods for finding the optimal stock S for the BC making use of the convexity of its cost function. In Appendix 7 the extensive results can be found.

To find out the performance of the method we look at the relative computation time difference as a measure, where T_{CE} is the computation time for using complete enumeration, where T_S is the computation time of the proposed *stop at first increase*-method, and where T_F is the computation time of the Fibonacci-method:

$$\Delta_S^{CE} = \frac{T^{CE} - T^S}{T^{CE}}$$

$$\Delta_F^{CE} = \frac{T^{CE} - T^F}{T^{CE}}$$

$$\Delta_S^{F} = \frac{T^{F} - T^S}{T^{F}}$$

Table 1: Summary of BC results

	min. (%)	mean (%)	median (%)	max. (%)
Δ_S^C	8.9	86.2	91.5	93.9
Δ_F^C	61.11	83.1	84.3	99.4
Δ_S^F	-1192.5	-11.2	4.6	29.6

Table 1 summarizes the relative performance of the search methods in the BC. The results indicate that the proof of convexity makes it possible to decrease the calculation time drastically.

4.2 Central Repair Shop

The computation time for the instances with N_1, N_2 being large, will be large, since the number of variables and equations in the system of equations of Section 3.2.2 increases. When we use complete enumeration we have to set an S^{max} to find the optimal number of spare machines. Since we have limited time, we chose to look at the results of Liang et al. (2013) and chose an S^{max} based on their findings. We decided that S_i^{max} should be constant for a given N_i. The values for S_i^{max} can be found below.

	(N_1, N_2)					
	$(10, 5)$	$(10, 10)$	$(10, 15)$	$(50, 25)$	$(50, 50)$	$(100, 50)$
S_1^{max}	20	20	20	25	25	25
S_2^{max}	16	20	22	25	25	31

To determine the performance of the several search methods in the CRS while using the Myopic(**R**) policy, we propose the following measures:

$$\Delta_{S1}^{CE} = \frac{T^{CE} - T^{S1}}{T^{CE}}$$

$$\Delta_{CD}^{CE} = \frac{T^{CE} - T^{F}}{T^{CE}}$$

$$\Delta_{CD}^{S1} = \frac{T^{S1} - T^{CD}}{T^{S1}}$$

Here is T_{CE} the computation time of finding the minimum while using complete enumeration again, T_{S1} the *stop at first increase*-method in one dimension and T_{CD} the coordinate descent method where the *stop at first increase*-method is used in every dimension.

Due to the time constraints not all results for the complete enumeration could be obtained. The last 40 instances are not created. In 44 out of 324 cases the same minimum as with the methods that did converge. These are therefore not included in the measures. For the other two methods all proposed instances are obtained and therefore included in Δ_{CD}^{S1}.

Table 2: Summary of CRS - Myopic(**R**) policy results

	min. (%)	mean (%)	median (%)	max. (%)
Δ_{S1}^{CE}	44.5	82.2	85.8	91.8
Δ_{CD}^{CE}	84.4	96.8	98.3	99.7
Δ_{CD}^{S1}	50.2	85.5	88.1	96.9

Table 2 summarizes the relative performance of the search methods in the BC. The results indicate that the two heuristics save a lot of computation time.

The *stop at first increase*-method in one direction converged in for all instances, which suggests that cost function for the CRS is also convex.

The coordinate descent method does not converge to the optimal stock S^* for all instances. In 44 out of 324 the method did not converge and higher costs were found. This is caused by the fact that in some cases no improvement of costs could be found in any direction i, but only in a direction where you adjust S_i and S_j with $i \neq j$ at the same time.

We only summarize the relative increase for the 44 instances that did not converge, and not include the instances for which the coordinate descent method did converge. The relative positive deviations from the minimum are summarized in Table 3.

Table 3: Deviation from the minimum costs

min. (%)	mean (%)	median (%)	max. (%)
1.7	0.4	0.2	1.7

The results indicate that the coordinate descent method does not deviate much from the minimum costs for the Myopic(**R**) policy and save a lot of computation time.

4.3 BC/CRS-equilibrium

To find the equilibria we used the *stop at first increase*-method, since it converges the fastest to the Myopic(**R**) optimum. Due to time constraints, and the long computation times, even for the *stop at first increase*-method, we did not compute all values of α. We did compute $alpha$ for $(N_1, N_2) \in \{(10,5),(10,10),(10,15),(50,25)\}$. The obtained values for α can be found in Appendix 7.

The average value of α equals 1.601. The results showed that their was a high correlation between the utilization factor u and the equilibrium factor α. The correlation between α and u equals 0.972. This is also shown in the scatter plot in Figure 3 and in Table 4 we give the mean values of α for different utilization factors:

Figure 3: Scatter plot showing the relation between utilization rate and equilibrium factor

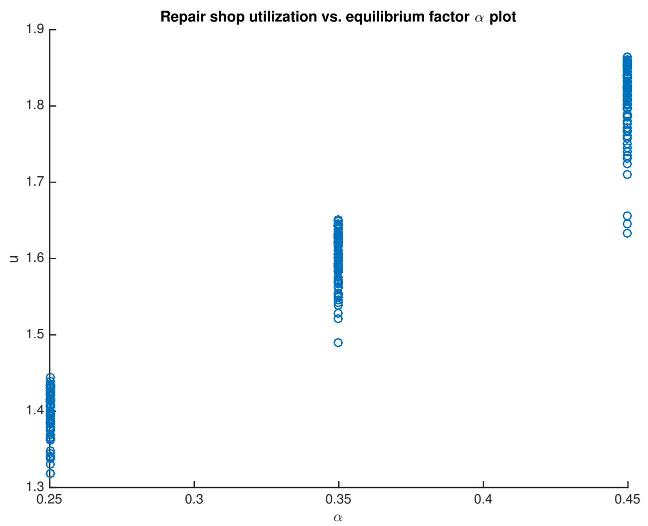


Table 4: Mean alpha for different utilization rates

u	0.25	0.35	0.45
$\bar{\alpha}$	1.399	1.601	1.803

5 Conclusion

We can conclude that due to the proof of convexity of the BC cost function smart search methods can be used to decrease the calculation time significantly for finding the minimum production line costs. The *stop at first increase*-method performs better than the Fibonacci search method in most of the cases, with a computation time decrease of respectively 86.2% and 83.1%. The better performance of the *stop at first increase*-method is mostly faster due to the fact that the optimal stock for the Myopic(**R**) policy is small for most instances and therefore this method does not use many iterations to find the minimum costs. Also, this method always converges to minimum costs. When a large stock is needed for to assure minimum costs, the Fibonacci method is preferred. The disadvantage of the Fibonacci method is that a maximum possible stock has to be set. This maximum number of spares is unknown, and therefore we do not know if the minimum will lie in the preset interval $[0, S^{max}]$. The *stop at first increase*-method is therefore preferred. If there would be a method to estimate the optimal stock, a hybrid method could be beneficial. Then the Fibonacci method will be used when the expected optimal stock is large.

For the Myopic(**R**) policy we proposed two heuristics to find the minimum costs. The first method will converge to the minimum if the cost function is convex, which seems very likely according the results, because in all the proposed instances the method converged. The average decrease in time is 82.2%. The second method we proposed has an even smaller average computation time (a decrease of 96.8 %), but does not converge in some cases (13.5%). The cost difference with the optimal value is on average 0.4% for the not converged cases. This means that the loss is really small and therefore the coordinate descent method has a great performance, since it saves lot of computation time. In Liang et al. (2013) an average loss of 0.66% was considered as close enough to the optimum. Due to this fast search method another 0.0005% is added, which can be considered as an insignificant increase.

Finding the equilibria where the BC has the same costs as the CRS with the Myopic(**R**) scheduling policy, has shown that the utilization rate is highly correlated with the factor α. We think this is caused by the fact that a high utilization makes combining repair shops less beneficial, and the assumption that a high factor α corresponds to the CRS being a less profitable substitute for the BC. Combining shops with a high utilization is less profitable since combining makes sense when the workforce in one shop can be used for repairing other machine types. If the workforce is already occupied all the time with repairing the machine type of its shop, it cannot be used for repairing other machines types.

6 Discussion and further research

Although improvements are made, there are several limitations to this research. Due to time constraints there was a lack code optimizing. This means that the computation times could have been lower, influencing the result. On the other hand, it is expected that this will reduce the computation times for all methods and that the relative difference will be approximately the same.

Also, not all instances were considered in this research due to the same time constraint as mentioned before. This means that we cannot use our recommendations for all instances.

For the complete enumeration, Fibonacci- and the *stop at first increase*-method in one direction for the CRS case, a maximum value for S had to be set before solving the problem. We did not find a way to find an upper bound for S such that convergence to the minimum is guaranteed. In further research it will be useful to find a way to set an S^{max} that assures convergence.

For the CRS we set S^{max} by looking at the results from Liang et al. (2013). This means that if we had to choose S^{max} without this knowledge, and we would have chosen another value, it could have been lower, meaning we would not find the minimum, or it could have been higher and the computation time difference in Section 4.1 and 4.2 would be even greater and the proposed *stop at first increase*-method in the BC and the coordinate descent method in the CRS are even more preferred.

Another thing that has to be looked in, is what happens when the utilization increases. The results suggest that the CRS will not be an improvement anymore when the single repair shops have a higher utilization, since the factor α might be greater than 2 when the utilization factor for the single repair shops increase.

References

Liang, W. K., Balcıoglu, B., and Svaluto, R. (2013). Scheduling policies for a repair shop problem. *Annals of Operations Research*, 211(1):273–288.

Sahba, P., Balcıoglu, B., and Banjevic, D. (2013). Spare parts provisioning for multiple *k*-out-of-*n*: G systems. *IIE transactions*, 45(9):953–963.

Taylor, J. and Jackson, R. (1954). An application of the birth and death process to the provision of spare machines. *OR*, 5(4):95–108.

7 Appendix

Table 5: The parameters and corresponding minimal costs for the BC and the Myopic(**R**) policy with the computation times of the several search methods and the equilibrium factor α.

PARAMETERS										Base Case						Myopic(**R**)					Myopic(**R**)				Equilibrium
N_1	N_2	λ_1	λ_2	μ_1	μ_2	h_1	h_2	b_1	b_2	Costs	S_1	S_2	T_{CE}	T_S	T_F	Costs	S_1	S_2	T_{CE}	T_S	Costs	S_1	S_2	T_{CD}	α
10	5	0.09	0.09	2	1	1	0.9	20	18	27.314	9	6	0.033	0.008	0.005	15.317	3	7	21.97	3.564	15.317	3	7	0.962	1.734
10	5	0.07	0.07	2	1	1	0.9	20	18	14.929	6	5	0.025	0.005	0.003	8.26	2	4	21.037	2.323	8.26	2	4	0.591	1.568
10	5	0.05	0.05	2	1	1	0.9	20	18	8.774	3	3	0.024	0.002	0.002	5.071	2	2	20.739	2.218	5.071	2	2	0.197	1.384
10	5	0.09	0.18	2	2	1	0.9	20	18	27.314	9	6	0.022	0.004	0.003	16.26	4	6	20.674	3.623	16.305	5	5	0.655	1.751
10	5	0.07	0.14	2	2	1	0.9	20	18	14.929	6	5	0.024	0.001	0.002	8.528	3	3	20.662	2.519	8.528	3	3	0.378	1.597
10	5	0.05	0.1	2	2	1	0.9	20	18	8.774	3	3	0.022	0.001	0.002	5.142	2	2	20.75	2.338	5.142	2	2	0.188	1.409
10	5	0.09	0.27	2	3	1	0.9	20	18	27.314	9	6	0.023	0.002	0.002	16.144	7	3	20.693	4.144	16.144	7	3	0.398	1.745
10	5	0.07	0.21	2	3	1	0.9	20	18	14.929	6	5	0.016	0.001	0.003	8.663	4	2	20.71	2.636	8.663	4	2	0.263	1.593
10	5	0.05	0.15	2	3	1	0.9	20	18	8.774	3	3	0.027	0.001	0.003	5.24	2	2	20.803	2.322	5.24	2	2	0.203	1.419
10	5	0.09	0.09	2	1	1	0.9	80	72	47.317	18	14	0.021	0.004	0.003	24.468	5	14	20.674	8.153	24.468	5	14	2.905	1.813
10	5	0.07	0.07	2	1	1	0.9	80	72	22.122	9	8	0.018	0.002	0.003	11.743	4	6	20.988	4.066	11.743	4	6	0.979	1.592
10	5	0.05	0.05	2	1	1	0.9	80	72	12.548	5	5	0.021	0.002	0.003	7.194	3	3	20.757	3.035	7.194	3	3	0.367	1.385
10	5	0.09	0.18	2	2	1	0.9	80	72	47.317	18	14	0.015	0.004	0.004	26.807	7	13	20.723	9.174	26.866	8	12	2.793	1.827
10	5	0.07	0.14	2	2	1	0.9	80	72	22.122	9	8	0.016	0.001	0.003	12.267	5	5	20.696	4.016	12.44	6	4	0.629	1.625
10	5	0.05	0.1	2	2	1	0.9	80	72	12.548	5	5	0.016	0.001	0.003	7.264	3	3	20.758	2.948	7.264	3	3	0.354	1.414
10	5	0.09	0.27	2	3	1	0.9	80	72	47.317	18	14	0.016	0.003	0.004	26.279	14	5	20.729	10.4	26.331	13	6	0.957	1.82
10	5	0.07	0.21	2	3	1	0.9	80	72	22.122	9	8	0.016	0.001	0.002	12.292	6	4	20.772	4.18	12.292	6	4	0.559	1.619
10	5	0.05	0.15	2	3	1	0.9	80	72	12.548	5	5	0.023	0.001	0.002	7.404	3	3	20.805	2.946	7.404	3	3	0.366	1.415
10	5	0.09	0.09	2	1	1	0.7	20	14	24.811	9	6	0.019	0.001	0.003	13.151	3	7	21.267	3.114	13.151	3	7	0.87	1.71
10	5	0.07	0.07	2	1	1	0.7	20	14	13.463	6	5	0.017	0.001	0.002	7.199	2	4	20.72	2.433	7.199	2	4	0.468	1.545
10	5	0.05	0.05	2	1	1	0.7	20	14	7.886	3	3	0.016	0.001	0.002	4.51	2	2	20.617	2.224	4.51	2	2	0.194	1.364
10	5	0.09	0.18	2	2	1	0.7	20	14	24.811	9	6	0.016	0.001	0.002	14.025	3	7	20.646	3.305	14.066	4	6	0.674	1.725
10	5	0.07	0.14	2	2	1	0.7	20	14	13.463	6	5	0.015	0.001	0.003	7.525	3	3	20.71	2.404	7.525	3	3	0.376	1.575
10	5	0.05	0.1	2	2	1	0.7	20	14	7.886	3	3	0.016	0.001	0.002	4.563	2	2	20.65	2.209	4.563	2	2	0.198	1.393
10	5	0.09	0.27	2	3	1	0.7	20	14	24.811	9	6	0.023	0.001	0.002	14.41	5	4	20.622	3.521	14.426	6	3	0.365	1.731
10	5	0.07	0.21	2	3	1	0.7	20	14	13.463	6	5	0.018	0.001	0.002	7.671	3	3	20.727	2.439	7.798	4	2	0.256	1.588
10	5	0.05	0.15	2	3	1	0.7	20	14	7.886	3	3	0.017	0.001	0.003	4.64	2	2	20.598	2.205	4.64	2	2	0.21	1.409
10	5	0.09	0.09	2	1	1	0.7	80	56	42.835	18	14	0.017	0.003	0.003	20.783	5	15	20.978	7.234	20.783	5	15	3.019	1.801
10	5	0.07	0.07	2	1	1	0.7	80	56	19.903	9	8	0.016	0.001	0.002	10.272	4	6	20.878	4.027	10.272	4	6	0.846	1.571
10	5	0.05	0.05	2	1	1	0.7	80	56	11.264	5	5	0.019	0.001	0.002	6.405	3	3	20.683	2.962	6.405	3	3	0.383	1.362
10	5	0.09	0.18	2	2	1	0.7	80	56	42.835	18	14	0.016	0.003	0.003	22.826	6	15	20.551	8.314	22.826	6	15	2.735	1.814
10	5	0.07	0.14	2	2	1	0.7	80	56	19.903	9	8	0.022	0.002	0.002	10.74	4	6	20.789	3.987	10.871	5	5	0.696	1.605
10	5	0.05	0.1	2	2	1	0.7	80	56	11.264	5	5	0.018	0.001	0.002	6.454	3	3	20.661	2.949	6.454	3	3	0.357	1.398
10	5	0.09	0.27	2	3	1	0.7	80	56	42.835	18	14	0.017	0.003	0.003	23.879	12	7	20.689	9.056	23.879	12	7	0.883	1.818
10	5	0.07	0.21	2	3	1	0.7	80	56	19.903	9	8	0.016	0.001	0.003	11.049	5	5	20.759	4.072	11.071	6	4	0.442	1.622
10	5	0.05	0.15	2	3	1	0.7	80	56	11.264	5	5	0.019	0.001	0.002	6.564	3	3	20.688	2.931	6.564	3	3	0.379	1.416
10	5	0.09	0.09	2	1	1	0.5	20	10	22.309	9	6	0.016	0.001	0.002	10.413	3	7	20.713	2.942	10.413	3	7	0.693	1.633
10	5	0.07	0.07	2	1	1	0.5	20	10	11.996	6	5	0.019	0.001	0.002	6.013	2	4	20.756	2.248	6.013	2	4	0.374	1.489
10	5	0.05	0.05	2	1	1	0.5	20	10	6.999	3	3	0.024	0.001	0.003	3.949	2	2	20.709	2.208	3.949	2	2	0.198	1.319
10	5	0.09	0.18	2	2	1	0.5	20	10	22.309	9	6	0.015	0.001	0.003	11.198	3	7	20.675	3.103	11.198	3	7	0.693	1.645
10	5	0.07	0.14	2	2	1	0.5	20	10	11.996	6	5	0.017	0.001	0.003	6.434	2	4	20.805	2.344	6.434	2	4	0.512	1.522
10	5	0.05	0.1	2	2	1	0.5	20	10	6.999	3	3	0.016	0	0.003	3.984	2	2	20.651	2.205	3.984	2	2	0.172	1.363
10	5	0.09	0.27	2	3	1	0.5	20	10	22.309	9	6	0.016	0.001	0.004	11.854	3	7	20.617	3.242	11.855	4	6	0.629	1.656
10	5	0.07	0.21	2	3	1	0.5	20	10	11.996	6	5	0.017	0.001	0.003	6.766	3	4	21.303	2.378	6.766	3	4	0.43	1.549
10	5	0.05	0.15	2	3	1	0.5	20	10	6.999	3	3	0.021	0.001	0.003	4.04	2	2	21.002	2.233	4.04	2	2	0.2	1.399
10	5	0.09	0.09	2	1	1	0.5	80	40	38.354	18	14	0.017	0.003	0.003	16.389	5	14	20.608	5.88	16.389	5	14	2.309	1.758
10	5	0.07	0.07	2	1	1	0.5	80	40	17.685	9	8	0.018	0.002	0.002	8.618	4	6	20.847	3.835	8.618	4	6	0.78	1.529
10	5	0.05	0.05	2	1	1	0.5	80	40	9.979	5	5	0.02	0.001	0.002	5.616	3	3	20.92	2.952	5.616	3	3	0.365	1.319
10	5	0.09	0.18	2	2	1	0.5	80	40	38.354	18	14	0.016	0.003	0.003	18.046	5	15	21.152	6.734	18.039	5	16	2.359	1.771
10	5	0.07	0.14	2	2	1	0.5	80	40	17.685	9	8	0.016	0.001	0.003	9.157	4	7	20.813	3.898	9.157	4	7	0.758	1.563
10	5	0.05	0.1	2	2	1	0.5	80	40	9.979	5	5	0.02	0.001	0.002	5.644	3	3	20.665	2.929	5.644	3	3	0.379	1.366
10	5	0.09	0.27	2	3	1	0.5	80	40	38.354	18	14	0.017	0.003	0.003	19.467	6	15	20.818	7.703	19.467	6	15	2.441	1.781
10	5	0.07	0.21	2	3	1	0.5	80	40	17.685	9	8	0.018	0.002	0.005	9.688	4	7	20.688	3.959	9.688	4	7	0.903	1.593
10	5	0.05	0.15	2	3	1	0.5	80	40	9.979	5	5	0.016	0.001	0.003	5.725	3	3	20.707	2.921	5.725	3	3	0.362	1.408
10	10	0.09	0.045	2	1	1	0.9	20	18	30.5	9	9	0.018	0.002	0.003	17.296	3	9	46.542	7.111	17.296	3	9	1.708	1.787
10	10	0.07	0.035	2	1	1	0.9	20	18	15.826	6	6	0.019	0.001	0.003	8.628	2	4	47.581	4.87	8.628	2	4	0.888	1.591
10	10	0.05	0.025	2	1	1	0.9	20	18	9.082	3	3	0.023	0.001	0.003	5.167	2	2	46.642	5.774	5.167	2	2	0.46	1.396
10	10	0.09	0.09	2	2	1	0.9	20	18	30.5	9	9	0.019	0.002	0.002	18.832	4	8	46.409	8.059	19.055	7	5	1.223	1.81
10	10	0.07	0.07	2	2	1	0.9	20	18	15.826	6	6	0.016	0.001	0.002	9.03	3	4	46.462	5.13	9.03	3	4	0.769	1.627
10	10	0.05	0.05	2	2	1	0.9	20	18	9.082	3	3	0.019	0.001	0.003	5.25	2	2	46.438	4.648	5.25	2	2	0.455	1.424
10	10	0.09	0.135	2	3	1	0.9	20	18	30.5	9	9	0.018	0.002	0.003	18.788	8	4	46.503	8.98	18.812	9	3	0.762	1.808
10	10	0.07	0.105	2	3	1	0.9	20	18	15.826	6	6	0.018	0.001	0.002	9.156	4	3	46.362	5.218	9.156	4	3	0.693	1.628
10	10	0.05	0.075	2	3	1	0.9	20	18	9.082	3	3	0.024	0.001	0.002	5.368	2	2	46.974	4.72	5.368	2	2	0.451	1.435

Table 6: The parameters and corresponding minimal costs for the BC and the Myopic(**R**) policy with the computation times of the several search methods and the equilibrium factor α.

PARAMETERS										Base Case						Myopic(**R**)					Myopic(**R**)				Equilibrium
N_1	N_2	λ_1	λ_2	μ_1	μ_2	h_1	h_2	b_1	b_2	Costs	S_1	S_2	T_{CE}	T_S	T_F	Costs	S_1	S_2	T_{CE}	T_S	Costs	S_1	S_2	T_{CD}	α
10	10	0.09	0.045	2	1	1	0.9	80	72	51.585	18	18	0.034	0.004	0.004	26.729	5	16	46.577	14.803	26.729	5	16	5.919	1.836
10	10	0.07	0.035	2	1	1	0.9	80	72	23.065	9	9	0.017	0.002	0.003	12.129	4	6	46.593	8.411	12.129	4	6	1.404	1.604
10	10	0.05	0.025	2	1	1	0.9	80	72	12.859	5	5	0.018	0.001	0.002	7.305	3	3	46.477	6.146	7.305	3	3	0.693	1.388
10	10	0.09	0.09	2	2	1	0.9	80	72	51.585	18	18	0.017	0.005	0.003	29.889	7	16	46.346	20.668	30.026	9	14	6.27	1.853
10	10	0.07	0.07	2	2	1	0.9	80	72	23.065	9	9	0.016	0.002	0.003	12.764	5	6	46.539	8.605	12.764	5	6	1.386	1.643
10	10	0.05	0.05	2	2	1	0.9	80	72	12.859	5	5	0.023	0.001	0.002	7.386	3	3	47.187	6.176	7.386	3	3	0.601	1.423
10	10	0.09	0.135	2	3	1	0.9	80	72	51.585	18	18	0.018	0.004	0.003	29.349	16	6	46.801	25.133	29.349	16	6	1.705	1.85
10	10	0.07	0.105	2	3	1	0.9	80	72	23.065	9	9	0.018	0.002	0.003	12.884	7	4	46.609	8.54	12.884	7	4	0.692	1.638
10	10	0.05	0.075	2	3	1	0.9	80	72	12.859	5	5	0.018	0.001	0.002	7.549	3	3	46.746	6.131	7.549	3	3	0.581	1.43
10	10	0.09	0.045	2	1	1	0.7	20	14	27.289	9	9	0.017	0.002	0.002	14.641	3	9	46.559	6.776	14.641	3	9	1.652	1.768
10	10	0.07	0.035	2	1	1	0.7	20	14	14.16	6	6	0.02	0.001	0.002	7.493	2	4	46.537	4.871	7.493	2	4	0.68	1.569
10	10	0.05	0.025	2	1	1	0.7	20	14	8.126	3	3	0.026	0.001	0.003	4.586	2	2	46.408	4.713	4.586	2	2	0.306	1.374
10	10	0.09	0.09	2	2	1	0.7	20	14	27.289	9	9	0.017	0.002	0.002	16.004	3	9	46.813	7.382	16.004	3	9	1.899	1.788
10	10	0.07	0.07	2	2	1	0.7	20	14	14.16	6	6	0.017	0.001	0.002	7.88	3	4	46.467	4.908	7.88	3	4	0.699	1.606
10	10	0.05	0.05	2	2	1	0.7	20	14	8.126	3	3	0.016	0.001	0.002	4.649	2	2	46.492	4.604	4.649	2	2	0.29	1.407
10	10	0.09	0.135	2	3	1	0.7	20	14	27.289	9	9	0.018	0.002	0.003	16.715	7	4	46.397	8.095	16.715	7	4	0.963	1.801
10	10	0.07	0.105	2	3	1	0.7	20	14	14.16	6	6	0.02	0.001	0.003	8.159	4	3	46.446	5.667	8.159	4	3	0.453	1.626
10	10	0.05	0.075	2	3	1	0.7	20	14	8.126	3	3	0.024	0.001	0.003	4.741	2	2	46.493	4.642	4.741	2	2	0.303	1.432
10	10	0.09	0.045	2	1	1	0.7	80	56	46.155	18	18	0.018	0.004	0.005	22.457	5	17	47.375	14.641	22.457	5	17	5.702	1.824
10	10	0.07	0.035	2	1	1	0.7	80	56	20.637	9	9	0.018	0.002	0.003	10.575	4	7	46.608	8.476	10.575	4	7	1.52	1.583
10	10	0.05	0.025	2	1	1	0.7	80	56	11.506	5	5	0.017	0.001	0.002	6.493	3	3	46.391	6.113	6.493	3	3	0.58	1.369
10	10	0.09	0.09	2	2	1	0.7	80	56	46.155	18	18	0.017	0.005	0.004	25.137	6	18	46.534	19.395	25.137	6	18	5.804	1.842
10	10	0.07	0.07	2	2	1	0.7	80	56	20.637	9	9	0.02	0.004	0.003	11.125	4	7	47.13	8.397	11.125	4	7	1.584	1.622
10	10	0.05	0.05	2	2	1	0.7	80	56	11.506	5	5	0.024	0.002	0.002	6.551	3	3	46.415	6.089	6.551	3	3	0.584	1.407
10	10	0.09	0.135	2	3	1	0.7	80	56	46.155	18	18	0.018	0.005	0.003	26.682	14	7	46.361	21.971	26.682	14	7	3.71	1.849
10	10	0.07	0.105	2	3	1	0.7	80	56	20.637	9	9	0.017	0.004	0.003	11.524	6	5	46.668	9.839	11.524	6	5	0.994	1.643
10	10	0.05	0.075	2	3	1	0.7	80	56	11.506	5	5	0.018	0.002	0.002	6.68	3	3	46.795	6.333	6.68	3	3	0.595	1.43
10	10	0.09	0.045	2	1	1	0.5	20	10	24.079	9	9	0.018	0.002	0.003	11.98	3	10	46.456	6.452	11.98	3	10	1.609	1.74
10	10	0.07	0.035	2	1	1	0.5	20	10	12.494	6	6	0.018	0.002	0.003	6.358	2	4	46.418	4.767	6.358	2	4	0.668	1.538
10	10	0.05	0.025	2	1	1	0.5	20	10	7.17	3	3	0.024	0.001	0.002	4.006	2	2	46.645	4.64	4.006	2	2	0.297	1.344
10	10	0.09	0.09	2	2	1	0.5	20	10	24.079	9	9	0.017	0.002	0.002	12.995	3	10	51.212	6.897	12.995	3	10	1.56	1.757
10	10	0.07	0.07	2	2	1	0.5	20	10	12.494	6	6	0.016	0.001	0.002	6.72	2	5	46.511	4.898	6.729	3	4	0.672	1.572
10	10	0.05	0.05	2	2	1	0.5	20	10	7.17	3	3	0.017	0.001	0.003	4.048	2	2	46.495	4.622	4.048	2	2	0.365	1.382
10	10	0.09	0.135	2	3	1	0.5	20	10	24.079	9	9	0.018	0.002	0.003	13.859	4	9	46.908	7.235	13.859	4	9	1.618	1.771
10	10	0.07	0.105	2	3	1	0.5	20	10	12.494	6	6	0.017	0.001	0.002	6.946	3	4	46.308	5.086	6.946	3	4	0.715	1.601
10	10	0.05	0.075	2	3	1	0.5	20	10	7.17	3	3	0.024	0.001	0.002	4.114	2	2	46.369	4.629	4.114	2	2	0.311	1.406
10	10	0.09	0.045	2	1	1	0.5	80	40	40.725	18	18	0.026	0.004	0.004	18.154	5	18	46.479	14.718	18.154	5	18	5.261	1.809
10	10	0.07	0.035	2	1	1	0.5	80	40	18.21	9	9	0.018	0.002	0.003	8.962	4	7	46.508	8.384	8.962	4	7	1.374	1.555
10	10	0.05	0.025	2	1	1	0.5	80	40	10.152	5	5	0.018	0.001	0.002	5.682	3	3	46.438	6.103	5.682	3	3	0.592	1.338
10	10	0.09	0.09	2	2	1	0.5	80	40	40.725	18	18	0.016	0.004	0.003	20.184	5	20	46.741	16.596	20.184	5	20	5.668	1.826
10	10	0.07	0.07	2	2	1	0.5	80	40	18.21	9	9	0.017	0.002	0.002	9.391	4	7	46.501	8.466	9.391	4	7	1.363	1.59
10	10	0.05	0.05	2	2	1	0.5	80	40	10.152	5	5	0.018	0.001	0.002	5.716	3	3	46.521	6.104	5.716	3	3	0.555	1.376
10	10	0.09	0.135	2	3	1	0.5	80	40	40.725	18	18	0.025	0.004	0.003	21.991	6	20	46.412	18.706	21.991	6	20	6.044	1.836
10	10	0.07	0.105	2	3	1	0.5	80	40	18.21	9	9	0.018	0.002	0.004	9.888	4	7	56.175	8.284	9.92	5	6	1.284	1.621
10	10	0.05	0.075	2	3	1	0.5	80	40	10.152	5	5	0.019	0.001	0.002	5.81	3	3	46.807	6.088	5.81	3	3	0.588	1.412
10	15	0.09	0.03	2	1	1	0.9	20	18	32.515	9	11	0.019	0.002	0.003	18.583	3	10	71.409	11.557	18.583	3	10	3.422	1.796
10	15	0.07	0.023	2	1	1	0.9	20	18	16.242	6	6	0.018	0.002	0.003	9.249	2	5	91.299	7.532	9.249	2	5	1.45	1.611
10	15	0.05	0.017	2	1	1	0.9	20	18	9.235	3	3	0.019	0.001	0.003	5.593	2	2	82.633	6.832	5.593	2	2	0.63	1.43
10	15	0.09	0.06	2	2	1	0.9	20	18	32.515	9	11	0.026	0.002	0.003	19.67	4	9	71.362	12.251	19.691	5	8	3.178	1.813
10	15	0.07	0.047	2	2	1	0.9	20	18	16.242	6	6	0.022	0.001	0.002	9.218	3	4	71.234	7.419	9.314	4	3	0.893	1.632
10	15	0.05	0.033	2	2	1	0.9	20	18	9.235	3	3	0.018	0.001	0.004	5.32	2	2	71.389	6.793	5.32	2	2	0.632	1.427
10	15	0.09	0.09	2	3	1	0.9	20	18	32.515	9	11	0.021	0.002	0.004	19.144	8	4	70.987	12.973	19.144	8	4	1.464	1.806
10	15	0.07	0.07	2	3	1	0.9	20	18	16.242	6	6	0.028	0.001	0.004	9.042	4	3	71.718	7.415	9.042	4	3	0.631	1.621
10	15	0.05	0.05	2	3	1	0.9	20	18	9.235	3	3	0.024	0.001	0.004	5.247	2	2	71.017	6.814	5.247	2	2	0.424	1.414
10	15	0.09	0.03	2	1	1	0.9	80	72	54.062	18	21	0.025	0.005	0.005	28.08	5	18	71.562	22.368	28.08	5	18	11.143	1.839
10	15	0.07	0.023	2	1	1	0.9	80	72	23.514	9	10	0.019	0.002	0.004	12.722	4	7	71.475	13.143	12.722	4	7	2.762	1.615
10	15	0.05	0.017	2	1	1	0.9	80	72	13.015	5	5	0.019	0.001	0.005	7.669	3	4	71.767	9.093	7.669	3	4	1.029	1.413
10	15	0.09	0.06	2	2	1	0.9	80	72	54.062	18	21	0.018	0.005	0.005	30.839	7	17	71.346	29.868	30.975	9	15	9.831	1.854
10	15	0.07	0.047	2	2	1	0.9	80	72	23.514	9	10	0.019	0.002	0.007	12.952	5	6	71.088	12.49	13.05	6	5	1.594	1.645
10	15	0.05	0.033	2	2	1	0.9	80	72	13.015	5	5	0.023	0.001	0.003	7.468	3	3	77.976	8.986	7.468	3	3	0.824	1.425
10	15	0.09	0.09	2	3	1	0.9	80	72	54.062	18	21	0.026	0.005	0.003	29.815	16	6	103.103	36.873	29.815	16	6	2.586	1.851
10	15	0.07	0.07	2	3	1	0.9	80	72	23.514	9	10	0.019	0.002	0.003	12.737	6	4	84.735	12.287	12.737	6	4	1.204	1.633
10	15	0.05	0.05	2	3	1	0.9	80	72	13.015	5	5	0.019	0.001	0.003	7.383	3	3	70.934	8.966	7.383	3	3	0.803	1.416

Table 7: The parameters and corresponding minimal costs for the BC and the Myopic(**R**) policy with the computation times of the several search methods and the equilibrium factor α.

PARAMETERS										Base Case						Myopic(**R**)					Myopic(**R**)				Equilibrium
N_1	N_2	λ_1	λ_2	μ_1	μ_2	h_1	h_2	b_1	b_2	Costs	S_1	S_2	T_{CE}	T_S	T_F	Costs	S_1	S_2	T_{CE}	T_S	Costs	S_1	S_2	T_{CD}	α
10	15	0.09	0.03	2	1	1	0.7	20	14	28.857	9	11	0.019	0.002	0.003	16.237	3	11	71.523	11.332	16.237	3	11	3.376	1.798
10	15	0.07	0.023	2	1	1	0.7	20	14	14.484	6	6	0.019	0.001	0.003	8.151	2	5	71.219	7.393	8.151	2	5	1.525	1.602
10	15	0.05	0.017	2	1	1	0.7	20	14	8.245	3	3	0.027	0.001	0.003	4.982	2	3	71.113	6.839	4.982	2	3	0.735	1.421
10	15	0.09	0.06	2	2	1	0.7	20	14	28.857	9	11	0.021	0.003	0.003	17.404	3	11	70.98	12.043	17.408	4	10	3.016	1.815
10	15	0.07	0.047	2	2	1	0.7	20	14	14.484	6	6	0.018	0.002	0.002	8.267	3	4	71.052	7.426	8.267	3	4	1.216	1.629
10	15	0.05	0.033	2	2	1	0.7	20	14	8.245	3	3	0.017	0.001	0.002	4.808	2	2	72.317	6.777	4.808	2	2	0.623	1.43
10	15	0.09	0.09	2	3	1	0.7	20	14	28.857	9	11	0.019	0.002	0.004	17.871	7	5	70.986	12.684	17.871	7	5	1.905	1.825
10	15	0.07	0.07	2	3	1	0.7	20	14	14.484	6	6	0.019	0.001	0.003	8.267	4	3	70.887	7.409	8.267	4	3	0.655	1.638
10	15	0.05	0.05	2	3	1	0.7	20	14	8.245	3	3	0.029	0.001	0.003	4.733	2	2	70.906	6.699	4.733	2	2	0.438	1.43
10	15	0.09	0.03	2	1	1	0.7	80	56	48.081	18	21	0.021	0.005	0.004	24.123	5	19	71.57	23.272	24.123	5	19	11.025	1.84
10	15	0.07	0.023	2	1	1	0.7	80	56	20.986	9	10	0.019	0.002	0.003	11.219	4	7	71.544	13.103	11.219	4	7	2.7	1.602
10	15	0.05	0.017	2	1	1	0.7	80	56	11.627	5	5	0.019	0.001	0.003	6.818	3	4	71.208	9.078	6.818	3	4	1.089	1.401
10	15	0.09	0.06	2	2	1	0.7	80	56	48.081	18	21	0.018	0.006	0.003	26.676	6	20	70.961	29.379	26.676	6	20	9.885	1.856
10	15	0.07	0.047	2	2	1	0.7	80	56	20.986	9	10	0.019	0.003	0.003	11.498	4	7	71.939	12.395	11.601	5	6	1.856	1.635
10	15	0.05	0.033	2	2	1	0.7	80	56	11.627	5	5	0.022	0.001	0.002	6.738	3	3	71.088	8.952	6.738	3	3	0.833	1.422
10	15	0.09	0.09	2	3	1	0.7	80	56	48.081	18	21	0.026	0.007	0.003	28.04	15	8	70.933	35.562	28.04	15	8	3.464	1.86
10	15	0.07	0.07	2	3	1	0.7	80	56	20.986	9	10	0.026	0.002	0.003	11.611	6	5	71.592	13.887	11.611	6	5	1.36	1.65
10	15	0.05	0.05	2	3	1	0.7	80	56	11.627	5	5	0.022	0.001	0.003	6.652	3	3	70.877	9.016	6.652	3	3	0.832	1.428
10	15	0.09	0.03	2	1	1	0.5	20	10	25.198	9	11	0.02	0.002	0.003	12.558	3	11	71.432	9.762	12.558	3	11	2.558	1.767
10	15	0.07	0.023	2	1	1	0.5	20	10	12.725	6	6	0.019	0.002	0.003	6.442	2	5	71.412	7.014	6.442	2	5	1.038	1.55
10	15	0.05	0.017	2	1	1	0.5	20	10	7.255	3	3	0.019	0.001	0.003	4.03	2	2	71.194	6.732	4.03	2	2	0.558	1.349
10	15	0.09	0.06	2	2	1	0.5	20	10	25.198	9	11	0.017	0.002	0.003	13.786	3	11	71.111	10.664	13.786	3	11	2.627	1.786
10	15	0.07	0.047	2	2	1	0.5	20	10	12.725	6	6	0.028	0.001	0.003	6.835	2	5	78.692	7.1	6.856	3	4	0.913	1.587
10	15	0.05	0.033	2	2	1	0.5	20	10	7.255	3	3	0.018	0.001	0.003	4.075	2	2	71.14	6.716	4.075	2	2	0.424	1.387
10	15	0.09	0.09	2	3	1	0.5	20	10	25.198	9	11	0.019	0.002	0.003	14.861	4	11	71.327	11.447	14.861	4	11	2.92	1.803
10	15	0.07	0.07	2	3	1	0.5	20	10	12.725	6	6	0.019	0.001	0.003	7.107	3	4	71.24	7.246	7.107	3	4	0.948	1.618
10	15	0.05	0.05	2	3	1	0.5	20	10	7.255	3	3	0.019	0.001	0.003	4.147	2	2	71.344	6.736	4.147	2	2	0.714	1.415
10	15	0.09	0.03	2	1	1	0.5	80	40	42.101	18	21	0.019	0.005	0.003	18.777	5	19	71.015	20.551	18.777	5	19	8.493	1.822
10	15	0.07	0.023	2	1	1	0.5	80	40	18.459	9	10	0.027	0.002	0.003	9.037	4	7	70.942	12.086	9.037	4	7	2.074	1.561
10	15	0.05	0.017	2	1	1	0.5	80	40	10.239	5	5	0.02	0.001	0.003	5.701	3	4	71.326	8.967	5.701	3	4	0.973	1.341
10	15	0.09	0.06	2	2	1	0.5	80	40	42.101	18	21	0.018	0.005	0.003	21.065	5	21	72.205	27.626	21.065	5	21	8.68	1.841
10	15	0.07	0.047	2	2	1	0.5	80	40	18.459	9	10	0.018	0.002	0.003	9.514	4	7	71.344	12.099	9.514	4	7	1.893	1.598
10	15	0.05	0.033	2	2	1	0.5	80	40	10.239	5	5	0.018	0.001	0.003	5.746	3	3	72.843	9.217	5.746	3	3	0.832	1.38
10	15	0.09	0.09	2	3	1	0.5	80	40	42.101	18	21	0.021	0.006	0.003	23.147	6	22	70.918	30.627	23.147	6	22	9.266	1.851
10	15	0.07	0.07	2	3	1	0.5	80	40	18.459	9	10	0.025	0.002	0.003	10.063	4	8	71.018	12.139	10.085	5	6	1.791	1.631
10	15	0.05	0.05	2	3	1	0.5	80	40	10.239	5	5	0.019	0.001	0.003	5.847	3	3	71.178	9.019	5.847	3	3	0.819	1.417
50	25	0.009	0.009	1	0.5	1	0.9	20	18	43.973	16	13	0.048	0.008	0.006	21.353	4	12	1453.104	234.25	21.353	4	12	51.142	1.814
50	25	0.007	0.007	1	0.5	1	0.9	20	18	18.27	7	7	0.045	0.004	0.005	9.274	3	4	1451.782	145.621	9.274	3	4	18.71	1.605
50	25	0.005	0.005	1	0.5	1	0.9	20	18	9.697	4	4	0.054	0.002	0.005	5.304	2	2	1459.374	137.172	5.304	2	2	8.348	1.398
50	25	0.009	0.018	1	1	1	0.9	20	18	43.973	16	13	0.037	0.008	0.006	23.879	5	12	1457.404	280.985	23.907	6	11	62.868	1.834
50	25	0.007	0.014	1	1	1	0.9	20	18	18.27	7	7	0.033	0.004	0.005	9.8	3	4	1453.618	148.786	9.8	3	4	17.843	1.643
50	25	0.005	0.01	1	1	1	0.9	20	18	9.697	4	4	0.035	0.004	0.005	5.408	2	2	1516.766	137.988	5.408	2	2	8.32	1.434
50	25	0.009	0.027	1	1.5	1	0.9	20	18	43.973	16	13	0.044	0.008	0.006	23.662	12	4	1460.257	326.783	23.662	12	4	22.054	1.83
50	25	0.007	0.021	1	1.5	1	0.9	20	18	18.27	7	7	0.035	0.004	0.006	9.941	5	3	1456.937	154.573	9.941	5	3	9.119	1.642
50	25	0.005	0.015	1	1.5	1	0.9	20	18	9.697	4	4	0.035	0.002	0.006	5.556	2	2	1431.981	138.098	5.556	2	2	8.297	1.445
50	25	0.009	0.009	1	0.5	1	0.9	80	72	67.511	28	24	0.037	0.022	0.006	31.08	5	21	1433.031	443.566	31.098	6	20	145.905	1.855
50	25	0.007	0.007	1	0.5	1	0.9	80	72	25.61	11	10	0.042	0.006	0.006	12.685	4	7	1432.654	236.732	12.685	4	7	33.678	1.61
50	25	0.005	0.005	1	0.5	1	0.9	80	72	13.495	6	6	0.034	0.004	0.005	7.463	3	3	1436.603	176.888	7.463	3	3	12.694	1.393
50	25	0.009	0.018	1	1	1	0.9	80	72	67.511	28	24	0.034	0.016	0.009	35.53	7	22	1440.878	587.464	35.602	9	20	186.174	1.864
50	25	0.007	0.014	1	1	1	0.9	80	72	25.61	11	10	0.044	0.006	0.008	13.518	5	6	1484.539	234.55	13.518	5	6	28.734	1.651
50	25	0.005	0.01	1	1	1	0.9	80	72	13.495	6	6	0.034	0.003	0.009	7.562	3	3	1431.683	176.895	7.562	3	3	10.892	1.429
50	25	0.009	0.027	1	1.5	1	0.9	80	72	67.511	28	24	0.034	0.015	0.007	34.727	21	6	1437.15	797.076	34.727	21	6	39.777	1.864
50	25	0.007	0.021	1	1.5	1	0.9	80	72	25.61	11	10	0.039	0.006	0.005	13.61	7	4	1430.218	238.872	13.61	7	4	16.122	1.646
50	25	0.005	0.015	1	1.5	1	0.9	80	72	13.495	6	6	0.042	0.003	0.006	7.76	4	3	1440.094	178.851	7.76	4	3	8.258	1.433
50	25	0.009	0.009	1	0.5	1	0.7	20	14	39.744	16	13	0.036	0.009	0.006	17.866	3	13	1430.713	223.365	17.866	3	13	52.834	1.797
50	25	0.007	0.007	1	0.5	1	0.7	20	14	16.4	7	7	0.034	0.004	0.005	8.055	2	5	1438.788	143.316	8.055	2	5	18.199	1.583
50	25	0.005	0.005	1	0.5	1	0.7	20	14	8.685	4	4	0.045	0.003	0.005	4.704	2	2	1436.616	137.492	4.704	2	2	8.291	1.377
50	25	0.009	0.018	1	1	1	0.7	20	14	39.744	16	13	0.041	0.008	0.006	20	4	13	1433.632	255.476	20	4	13	57.264	1.814
50	25	0.007	0.014	1	1	1	0.7	20	14	16.4	7	7	0.036	0.004	0.005	8.505	3	5	1454.179	147.309	8.505	3	5	18.089	1.621
50	25	0.005	0.01	1	1	1	0.7	20	14	8.685	4	4	0.034	0.002	0.006	4.782	2	2	1454.054	137.578	4.782	2	2	8.381	1.414
50	25	0.009	0.027	1	1.5	1	0.7	20	14	39.744	16	13	0.046	0.009	0.006	21.267	10	5	1458.475	298.332	21.267	10	5	26.851	1.825
50	25	0.007	0.021	1	1.5	1	0.7	20	14	16.4	7	7	0.066	0.005	0.006	8.84	4	3	1514.793	151.639	8.84	4	3	11.655	1.642
50	25	0.005	0.015	1	1.5	1	0.7	20	14	8.685	4	4	0.034	0.004	0.006	4.898	2	2	1504.906	136.928	4.898	2	2	8.246	1.439

Table 8: The parameters and corresponding minimal costs for the BC and the Myopic(**R**) policy with the computation times of the several search methods and the equilibrium factor α.

PARAMETERS										Base Case						Myopic(**R**)					Myopic(**R**)				Equilibrium
N_1	N_2	λ_1	λ_2	μ_1	μ_2	h_1	h_2	b_1	b_2	Costs	S_1	S_2	T_{CE}	T_S	T_F	Costs	S_1	S_2	T_{CE}	T_S	Costs	S_1	S_2	T_{CD}	α
50	25	0.009	0.009	1	0.5	1	0.7	80	56	60.854	28	24	0.048	0.019	0.007	25.84	5	21	1456.349	423.095	25.84	5	21	142.7	1.845
50	25	0.007	0.007	1	0.5	1	0.7	80	56	22.969	11	10	0.095	0.006	0.006	10.999	4	7	1459.665	232.717	10.999	4	7	30.486	1.589
50	25	0.005	0.005	1	0.5	1	0.7	80	56	12.083	6	6	0.061	0.004	0.005	6.629	3	3	1457.093	176.107	6.629	3	3	12.312	1.37
50	25	0.009	0.018	1	1	1	0.7	80	56	60.854	28	24	0.033	0.017	0.009	29.549	6	24	1452.622	613.526	29.549	6	24	153.57	1.857
50	25	0.007	0.014	1	1	1	0.7	80	56	22.969	11	10	0.035	0.009	0.008	11.785	4	7	1452.176	231.964	11.785	4	7	34.537	1.629
50	25	0.005	0.01	1	1	1	0.7	80	56	12.083	6	6	0.04	0.005	0.009	6.7	3	3	1459.618	177.934	6.7	3	3	10.615	1.413
50	25	0.009	0.027	1	1.5	1	0.7	80	56	60.854	28	24	0.033	0.016	0.01	31.833	19	7	1453.468	682.569	31.833	19	7	38.698	1.859
50	25	0.007	0.021	1	1.5	1	0.7	80	56	22.969	11	10	0.036	0.006	0.005	12.201	6	5	1452.33	234.554	12.201	6	5	21.374	1.651
50	25	0.005	0.015	1	1.5	1	0.7	80	56	12.083	6	6	0.04	0.003	0.007	6.858	3	3	1481.335	176.513	6.858	3	3	10.536	1.434
50	25	0.009	0.009	1	0.5	1	0.5	20	10	35.516	16	13	0.041	0.008	0.006	13.928	3	12	1451.736	198.018	13.928	3	12	40.665	1.761
50	25	0.007	0.007	1	0.5	1	0.5	20	10	14.53	7	7	0.033	0.004	0.005	6.689	2	5	1459.158	141.212	6.689	2	5	16.323	1.542
50	25	0.005	0.005	1	0.5	1	0.5	20	10	7.673	4	4	0.034	0.003	0.005	4.104	2	2	1453.82	138.073	4.104	2	2	5.441	1.338
50	25	0.009	0.018	1	1	1	0.5	20	10	35.516	16	13	0.042	0.01	0.006	15.746	4	13	1459.363	230.176	15.746	4	13	42.916	1.776
50	25	0.007	0.014	1	1	1	0.5	20	10	14.53	7	7	0.045	0.006	0.005	7.23	3	5	1450.82	166.333	7.23	3	5	15.353	1.583
50	25	0.005	0.01	1	1	1	0.5	20	10	7.673	4	4	0.033	0.004	0.006	4.157	2	2	1457.846	156.306	4.157	2	2	5.446	1.389
50	25	0.009	0.027	1	1.5	1	0.5	20	10	35.516	16	13	0.033	0.013	0.006	17.29	4	14	1508.839	271.811	17.334	5	13	47.306	1.789
50	25	0.007	0.021	1	1.5	1	0.5	20	10	14.53	7	7	0.046	0.004	0.005	7.755	3	5	1452.977	153.14	7.755	3	5	18.104	1.617
50	25	0.005	0.015	1	1.5	1	0.5	20	10	7.673	4	4	0.034	0.003	0.005	4.241	2	2	1453.584	137.081	4.241	2	2	5.444	1.435
50	25	0.009	0.009	1	0.5	1	0.5	80	40	54.196	28	24	0.038	0.015	0.007	20.146	5	21	1452.221	414.991	20.146	5	21	122.768	1.824
50	25	0.007	0.007	1	0.5	1	0.5	80	40	20.329	11	10	0.039	0.006	0.006	9.185	4	7	1461.705	227.969	9.185	4	7	30.936	1.553
50	25	0.005	0.005	1	0.5	1	0.5	80	40	10.671	6	6	0.034	0.004	0.005	5.752	3	4	1459.107	175.424	5.752	3	4	12.632	1.331
50	25	0.009	0.018	1	1	1	0.5	80	40	54.196	28	24	0.04	0.016	0.007	23.116	6	24	1451.773	519.428	23.116	6	24	122.759	1.839
50	25	0.007	0.014	1	1	1	0.5	80	40	20.329	11	10	0.05	0.006	0.008	9.928	4	8	1476.385	229.93	9.928	4	8	28.347	1.593
50	25	0.005	0.01	1	1	1	0.5	80	40	10.671	6	6	0.065	0.004	0.008	5.837	3	4	1452.765	175.887	5.837	3	4	12.548	1.384
50	25	0.009	0.027	1	1.5	1	0.5	80	40	54.196	28	24	0.038	0.018	0.009	25.83	7	25	1455.059	586.312	25.83	7	25	144.192	1.845
50	25	0.007	0.021	1	1.5	1	0.5	80	40	20.329	11	10	0.033	0.016	0.006	10.748	5	8	1458.207	234.754	10.91	6	6	23.814	1.629
50	25	0.005	0.015	1	1.5	1	0.5	80	40	10.671	6	6	0.045	0.004	0.006	5.954	3	3	1450.761	175.873	5.954	3	3	10.551	1.433
50	50	0.009	0.005	1	0.5	1	0.9	20	18	47.393	16	16	0.04	0.017	0.007	23.176	4	14	7484.042	1288.705	23.176	4	14	292.177	NA
50	50	0.007	0.004	1	0.5	1	0.9	20	18	18.723	7	7	0.041	0.005	0.007	9.436	3	5	7437.687	685.794	9.436	3	5	61.413	NA
50	50	0.005	0.003	1	0.5	1	0.9	20	18	9.772	4	4	0.05	0.003	0.006	5.335	2	2	7408.392	661.706	5.335	2	2	31.786	NA
50	50	0.009	0.009	1	1	1	0.9	20	18	47.393	16	16	0.04	0.011	0.007	26.485	5	14	7407.7	1511.692	26.485	5	14	408.325	NA
50	50	0.007	0.007	1	1	1	0.9	20	18	18.723	7	7	0.044	0.009	0.007	10.029	4	4	7407.102	719.242	10.029	4	4	56.809	NA
50	50	0.005	0.005	1	1	1	0.9	20	18	9.772	4	4	0.046	0.004	0.006	5.446	2	2	7410.266	653.533	5.446	2	2	31.966	NA
50	50	0.009	0.014	1	1.5	1	0.9	20	18	47.393	16	16	0.04	0.014	0.007	26.308	15	4	7478.192	1790.052	26.308	15	4	116.756	NA
50	50	0.007	0.011	1	1.5	1	0.9	20	18	18.723	7	7	0.049	0.005	0.007	10.173	5	3	7395.971	730.229	10.173	5	3	48.369	NA
50	50	0.005	0.008	1	1.5	1	0.9	20	18	9.772	4	4	0.041	0.003	0.007	5.604	2	2	7474.047	656.406	5.604	2	2	32.081	NA
50	50	0.009	0.005	1	0.5	1	0.9	80	72	71.349	28	28	0.042	0.021	0.012	32.992	5	23	7453.788	2185.062	33.007	6	22	845.849	NA
50	50	0.007	0.004	1	0.5	1	0.9	80	72	26.082	11	11	0.048	0.008	0.007	12.839	4	7	7103.699	1133.634	12.839	4	7	152.886	NA
50	50	0.005	0.003	1	0.5	1	0.9	80	72	13.57	6	6	0.039	0.004	0.006	7.499	3	3	7159.275	846.44	7.499	3	3	44.434	NA
50	50	0.009	0.009	1	1	1	0.9	80	72	71.349	28	28	0.046	0.024	0.011	38.299	7	25	7133.128	3003.015	38.364	9	23	1012.568	NA
50	50	0.007	0.007	1	1	1	0.9	80	72	26.082	11	11	0.047	0.017	0.011	13.729	5	7	7198.31	1130.485	13.778	6	6	117.487	NA
50	50	0.005	0.005	1	1	1	0.9	80	72	13.57	6	6	0.041	0.008	0.008	7.604	3	3	7127.738	842.757	7.604	3	3	38.929	NA
50	50	0.009	0.014	1	1.5	1	0.9	80	72	71.349	28	28	0.051	0.048	0.011	37.496	24	6	7218.52	3665.412	37.496	24	6	226.236	NA
50	50	0.007	0.011	1	1.5	1	0.9	80	72	26.082	11	11	0.04	0.01	0.007	13.876	7	5	7163.141	1144.718	13.876	7	5	97.067	NA
50	50	0.005	0.008	1	1.5	1	0.9	80	72	13.57	6	6	0.041	0.004	0.007	7.787	4	3	7123.998	846.887	7.787	4	3	36.177	NA
50	50	0.009	0.005	1	0.5	1	0.7	20	14	42.405	16	16	0.062	0.011	0.007	19.261	3	14	7196.873	1171.896	19.261	3	14	267.105	NA
50	50	0.007	0.004	1	0.5	1	0.7	20	14	16.752	7	7	0.052	0.005	0.006	8.173	2	5	7126.786	697.168	8.173	2	5	72.143	NA
50	50	0.005	0.003	1	0.5	1	0.7	20	14	8.743	4	4	0.04	0.003	0.007	4.729	2	2	7150.837	670.584	4.729	2	2	32.441	NA
50	50	0.009	0.009	1	1	1	0.7	20	14	42.405	16	16	0.047	0.015	0.009	21.985	4	16	7180.545	1396.222	21.985	4	16	293.744	NA
50	50	0.007	0.007	1	1	1	0.7	20	14	16.752	7	7	0.043	0.009	0.007	8.663	3	5	7235.986	714.165	8.663	3	5	71.253	NA
50	50	0.005	0.005	1	1	1	0.7	20	14	8.743	4	4	0.043	0.005	0.006	4.812	2	2	7151.04	669.145	4.812	2	2	32.054	NA
50	50	0.009	0.014	1	1.5	1	0.7	20	14	42.405	16	16	0.051	0.018	0.007	23.714	12	6	7163.711	1659.98	23.739	13	5	110.147	NA
50	50	0.007	0.011	1	1.5	1	0.7	20	14	16.752	7	7	0.04	0.007	0.007	9.076	4	4	7151.416	728.684	9.076	4	4	48.279	NA
50	50	0.005	0.008	1	1.5	1	0.7	20	14	8.743	4	4	0.043	0.005	0.006	4.936	2	2	7147.824	672.657	4.936	2	2	32.03	NA
50	50	0.009	0.005	1	0.5	1	0.5	20	10	37.416	16	16	0.04	0.013	0.007	15.275	3	15	7394.873	1042.344	15.275	3	15	228.07	NA
50	50	0.007	0.004	1	0.5	1	0.5	20	10	14.781	7	7	0.045	0.005	0.007	6.865	2	5	7432.689	691.068	6.865	2	5	64.997	NA
50	50	0.005	0.003	1	0.5	1	0.5	20	10	7.715	4	4	0.04	0.004	0.007	4.123	2	2	7435.399	692.175	4.123	2	2	21.595	NA
50	50	0.009	0.009	1	1	1	0.5	20	10	37.416	16	16	0.052	0.012	0.007	17.364	4	16	7412.284	1217.044	17.364	4	16	235.591	NA
50	50	0.007	0.007	1	1	1	0.5	20	10	14.781	7	7	0.061	0.005	0.006	7.289	3	5	7442.23	701.558	7.289	3	5	57.658	NA
50	50	0.005	0.005	1	1	1	0.5	20	10	7.715	4	4	0.042	0.004	0.007	4.179	2	2	7403.969	666.067	4.179	2	2	21.59	NA
50	50	0.009	0.014	1	1.5	1	0.5	20	10	37.416	16	16	0.061	0.013	0.007	19.213	4	18	7229.082	1396.555	19.244	5	16	272.762	NA
50	50	0.007	0.011	1	1.5	1	0.5	20	10	14.781	7	7	0.051	0.01	0.006	7.695	3	5	7345.35	716.754	7.695	3	5	63.033	NA
50	50	0.005	0.008	1	1.5	1	0.5	20	10	7.715	4	4	0.047	0.005	0.009	4.269	2	2	7587.133	686.645	4.269	2	2	21.596	NA

Table 9: The parameters and corresponding minimal costs for the BC and the Myopic(**R**) policy with the computation times of the several search methods and the equilibrium factor α.

PARAMETERS										Base Case						Myopic(**R**)					Myopic(**R**)				Equilibrium
N_1	N_2	λ_1	λ_2	μ_1	μ_2	h_1	h_2	b_1	b_2	Costs	S_1	S_2	T_{CE}	T_S	T_F	Costs	S_1	S_2	T_{CE}	T_S	Costs	S_1	S_2	T_{CD}	α
50	50	0.009	0.005	1	0.5	1	0.5	80	40	56.329	28	28	0.044	0.03	0.011	21.547	5	24	7543.511	2121.188	21.547	5	24	657.395	NA
50	50	0.007	0.004	1	0.5	1	0.5	80	40	20.591	11	11	0.04	0.011	0.01	9.404	4	7	7542.011	1124.063	9.404	4	7	113.569	NA
50	50	0.005	0.003	1	0.5	1	0.5	80	40	10.713	6	6	0.048	0.005	0.01	5.76	3	4	7563.225	871.006	5.76	3	4	50.962	NA
50	50	0.009	0.009	1	1	1	0.5	80	40	56.329	28	28	0.042	0.022	0.012	24.835	6	27	7563.054	2672.578	24.835	6	27	665.137	NA
50	50	0.007	0.007	1	1	1	0.5	80	40	20.591	11	11	0.04	0.009	0.007	9.983	4	8	7574.397	1125.408	9.983	4	8	139.936	NA
50	50	0.005	0.005	1	1	1	0.5	80	40	10.713	6	6	0.051	0.005	0.007	5.848	3	4	7543.598	860.553	5.848	3	4	48.977	NA
50	50	0.009	0.014	1	1.5	1	0.5	80	40	56.329	28	28	0.04	0.024	0.008	27.885	6	31	7626.718	3280.902	27.901	7	29	748.54	NA
50	50	0.007	0.011	1	1.5	1	0.5	80	40	20.591	11	11	0.041	0.009	0.007	10.645	5	7	7555.343	1134.423	10.645	5	7	118.534	NA
50	50	0.005	0.008	1	1.5	1	0.5	80	40	10.713	6	6	0.05	0.004	0.006	5.98	3	4	7563.568	868.145	5.98	3	4	44.84	NA
100	50	0.009	0.009	2	1	1	0.9	20	18	50.85	19	16	0.05	0.021	0.009	23.463	4	14	34139.736	7289.758	23.463	4	14	1406.957	NA
100	50	0.007	0.007	2	1	1	0.9	20	18	19.111	7	7	0.062	0.01	0.009	9.467	3	5	34131.012	4285.679	9.467	3	5	384.294	NA
100	50	0.005	0.005	2	1	1	0.9	20	18	9.822	4	4	0.05	0.006	0.008	5.344	2	2	34318.437	4045.347	5.344	2	2	206.195	NA
100	50	0.009	0.018	2	2	1	0.9	20	18	50.85	19	16	0.056	0.018	0.008	26.825	5	15	34655.968	8650.188	26.829	6	14	1905.323	NA
100	50	0.007	0.014	2	2	1	0.9	20	18	19.111	7	7	0.052	0.008	0.009	10.068	4	4	33440.684	4322.545	10.068	4	4	346.776	NA
100	50	0.005	0.01	2	2	1	0.9	20	18	9.822	4	4	0.048	0.005	0.01	5.456	2	2	33427.998	4001.083	5.456	2	2	206.648	NA
100	50	0.009	0.027	2	3	1	0.9	20	18	50.85	19	16	0.086	0.021	0.012	26.597	15	4	33876.964	10024.321	26.597	15	4	560.754	NA
100	50	0.007	0.021	2	3	1	0.9	20	18	19.111	7	7	0.05	0.009	0.01	10.212	5	3	33777.683	4429.782	10.212	5	3	293.285	NA
100	50	0.005	0.015	2	3	1	0.9	20	18	9.822	4	4	0.065	0.005	0.012	5.616	2	2	33861.641	4049.498	5.616	2	2	205.682	NA
100	50	0.009	0.009	2	1	1	0.9	80	72	75.062	32	28	0.051	0.039	0.011	33.291	6	23	33894.042	11625.762	33.291	6	23	3611.345	NA
100	50	0.007	0.007	2	1	1	0.9	80	72	26.46	11	11	0.059	0.011	0.01	12.881	4	7	34509.803	6792.306	12.881	4	7	769.067	NA
100	50	0.005	0.005	2	1	1	0.9	80	72	13.62	6	6	0.053	0.007	0.008	7.51	3	3	33926.595	5170.887	7.51	3	3	284.681	NA
100	50	0.009	0.018	2	2	1	0.9	80	72	75.062	32	28	0.048	0.035	0.011	38.667	8	25	33807.278	15848.564	38.667	8	25	5117.674	NA
100	50	0.007	0.014	2	2	1	0.9	80	72	26.46	11	11	0.06	0.013	0.009	13.77	5	7	33960.574	6704.696	13.816	6	6	600.709	NA
100	50	0.005	0.01	2	2	1	0.9	80	72	13.62	6	6	0.048	0.01	0.009	7.615	3	3	NA	5129.143	7.615	3	3	250.007	NA
100	50	0.009	0.027	2	3	1	0.9	80	72	75.062	32	28	0.06	0.037	0.01	37.81	24	6	NA	18458.255	37.81	24	6	923.979	NA
100	50	0.007	0.021	2	3	1	0.9	80	72	26.46	11	11	0.06	0.011	0.01	13.915	7	5	NA	6473.239	13.915	7	5	474.948	NA
100	50	0.005	0.015	2	3	1	0.9	80	72	13.62	6	6	0.052	0.006	0.01	7.793	4	3	NA	4894.465	7.793	4	3	201.787	NA
100	50	0.009	0.009	2	1	1	0.7	20	14	45.861	19	16	0.069	0.019	0.014	19.53	3	15	NA	6335.898	19.533	4	14	1151.379	NA
100	50	0.007	0.007	2	1	1	0.7	20	14	17.14	7	7	0.054	0.008	0.012	8.208	3	5	NA	4002.047	8.208	3	5	383.28	NA
100	50	0.005	0.005	2	1	1	0.7	20	14	8.793	4	4	0.056	0.005	0.013	4.737	2	2	NA	3830.791	4.737	2	2	196.61	NA
100	50	0.009	0.018	2	2	1	0.7	20	14	45.861	19	16	0.052	0.016	0.009	22.283	4	16	NA	7456.536	22.283	4	16	1515.413	NA
100	50	0.007	0.014	2	2	1	0.7	20	14	17.14	7	7	0.056	0.008	0.009	8.702	3	5	NA	4092.109	8.702	3	5	423.573	NA
100	50	0.005	0.01	2	2	1	0.7	20	14	8.793	4	4	0.068	0.004	0.008	4.822	2	2	NA	3912.275	4.822	2	2	198.111	NA
100	50	0.009	0.027	2	3	1	0.7	20	14	45.861	19	16	0.066	0.027	0.009	24.01	12	6	NA	8744.257	24.021	13	5	515.749	NA
100	50	0.007	0.021	2	3	1	0.7	20	14	17.14	7	7	0.055	0.022	0.009	9.116	4	4	NA	4207.436	9.116	4	4	299.555	NA
100	50	0.005	0.015	2	3	1	0.7	20	14	8.793	4	4	0.053	0.005	0.009	4.947	2	2	NA	3816.172	4.947	2	2	196.483	NA
100	50	0.009	0.009	2	1	1	0.7	80	56	67.552	32	28	0.065	0.03	0.011	27.551	5	24	NA	11000.213	27.551	5	24	3328.369	NA
100	50	0.007	0.007	2	1	1	0.7	80	56	23.714	11	11	0.05	0.011	0.011	11.159	4	7	NA	6326.519	11.159	4	7	664.514	NA
100	50	0.005	0.005	2	1	1	0.7	80	56	12.192	6	6	0.058	0.006	0.009	6.659	3	4	NA	4879.284	6.659	3	4	309.986	NA
100	50	0.009	0.018	2	2	1	0.7	80	56	67.552	32	28	0.047	0.032	0.012	31.943	6	27	NA	14181.968	31.943	6	27	3626.5	NA
100	50	0.007	0.014	2	2	1	0.7	80	56	23.714	11	11	0.051	0.014	0.012	11.988	5	7	NA	6330.896	11.988	5	7	635.589	NA
100	50	0.005	0.01	2	2	1	0.7	80	56	12.192	6	6	0.062	0.006	0.012	6.744	3	3	NA	4865.622	6.744	3	3	238.861	NA
100	50	0.009	0.027	2	3	1	0.7	80	56	67.552	32	28	0.05	0.033	0.012	34.776	21	8	NA	17243.773	34.79	22	7	1016.454	NA
100	50	0.007	0.021	2	3	1	0.7	80	56	23.714	11	11	0.063	0.011	0.01	12.497	6	6	NA	6381.68	12.501	7	5	389.848	NA
100	50	0.005	0.015	2	3	1	0.7	80	56	12.192	6	6	0.052	0.006	0.009	6.912	3	3	NA	4914.757	6.912	3	3	236.097	NA
100	50	0.009	0.009	2	1	1	0.5	20	10	40.872	19	16	0.052	0.018	0.009	15.163	3	15	NA	5790.198	15.163	3	15	1109.568	NA
100	50	0.007	0.007	2	1	1	0.5	20	10	15.17	7	7	0.059	0.007	0.012	6.82	2	5	NA	3934.552	6.82	2	5	388.14	NA
100	50	0.005	0.005	2	1	1	0.5	20	10	7.764	4	4	0.05	0.004	0.009	4.131	2	2	NA	3810.101	4.131	2	2	134.207	NA
100	50	0.009	0.018	2	2	1	0.5	20	10	40.872	19	16	0.065	0.024	0.01	17.419	4	16	NA	6644.105	17.419	4	16	1191.978	NA
100	50	0.007	0.014	2	2	1	0.5	20	10	15.17	7	7	0.049	0.008	0.01	7.404	3	5	NA	4060.736	7.404	3	5	339.79	NA
100	50	0.005	0.01	2	2	1	0.5	20	10	7.764	4	4	0.082	0.004	0.014	4.188	2	2	NA	3806.069	4.188	2	2	134.052	NA
100	50	0.009	0.027	2	3	1	0.5	20	10	40.872	19	16	0.068	0.017	0.013	19.482	4	18	NA	7753.176	19.486	5	17	1389.161	NA
100	50	0.007	0.021	2	3	1	0.5	20	10	15.17	7	7	0.068	0.007	0.012	8.001	3	6	NA	4204.113	8.001	3	6	410.312	NA
100	50	0.005	0.015	2	3	1	0.5	20	10	7.764	4	4	0.051	0.005	0.012	4.278	2	2	NA	3825.521	4.278	2	2	134.171	NA
100	50	0.009	0.009	2	1	1	0.5	80	40	60.041	32	28	0.062	0.029	0.012	21.407	5	23	NA	10130.069	21.407	5	23	2771.842	NA
100	50	0.007	0.007	2	1	1	0.5	80	40	20.969	11	11	0.05	0.01	0.014	9.31	4	7	NA	6211.038	9.31	4	7	577.067	NA
100	50	0.005	0.005	2	1	1	0.5	80	40	10.763	6	6	0.06	0.006	0.012	5.766	3	4	NA	4931.183	5.766	3	4	309.913	NA
100	50	0.009	0.018	2	2	1	0.5	80	40	60.041	32	28	0.048	0.038	0.01	24.874	6	27	NA	13109.551	24.874	6	27	2898.722	NA
100	50	0.007	0.014	2	2	1	0.5	80	40	20.969	11	11	0.049	0.01	0.008	10.105	4	8	NA	6261.44	10.105	4	8	617.324	NA
100	50	0.005	0.01	2	2	1	0.5	80	40	10.763	6	6	0.058	0.006	0.008	5.855	3	4	NA	4890.508	5.855	3	4	274.42	NA
100	50	0.009	0.027	2	3	1	0.5	80	40	60.041	32	28	0.049	0.028	0.011	28.14	7	30	NA	16330.898	28.14	7	30	3547.839	NA
100	50	0.007	0.021	2	3	1	0.5	80	40	20.969	11	11	0.063	0.011	0.009	10.964	5	8	NA	6339.321	10.964	5	8	624.86	NA
100	50	0.005	0.015	2	3	1	0.5	80	40	10.763	6	6	0.055	0.006	0.009	5.988	3	4	NA	4928.95	5.988	3	4	274.351	NA