Assignment 2

Oskar Pauli, Elias Sörstrand

May 17, 2021

1 Introduction

In this assignment we were given a mathematical model for a maintenance schedule that aims to minimize the cost of maintaining a system during a limited time period, while still making sure that everything is fully functioning. The system consists of several different parts together with costs of maintaining it as well as how long it lives before it needs maintenance. The aim of this report is to explain why and describe how our model reacts to different constraints and variants of them, firstly from how the optimal value changes, but also how the computation time varies.

2 Model

2.1 Model 1

Parameters used in the model:

- $\mathcal{N}$ = [1,10], the set of components in the system.
- T = the number of time steps in the planning period.
- $\Omega = \{(s,t)|0 \leq s < t \leq T+1; s,t \in \mathbb{Z}\}$, which is the set of replacement intervals.
- $c_{it} = [34\ 25\ 14\ 21\ 16\ 3\ 10\ 5\ 7\ 10]^T\ \mathbf{I}$, the cost of a new component $i \in \mathcal{N}$, $t \in 1,...,T$, with $\mathbf{I}$ as the $T \times T$. one-matrix.

Variables used in the model:

- $x_{it} = \begin{cases} 1 \text{ if component } i \in \mathcal{N} \text{ is replaced at time } t \in 1,...,T, \\ 0 \text{ otherwise.} \end{cases}$
- $z_t = \begin{cases} 1 \text{ if maintenance is made at time } t \in 1,...,T, \\ 0 \text{ otherwise.} \end{cases}$

The objective function in the model we wish to minimize is

$$min \quad z = \sum_{t=1}^{T} \left(\sum_{i=1}^{10} c_{it} x_{it} + d_t z_t \right). \tag{1}$$

The reason we are using this objective function, equation 1, is since we want to minimize the cost of maintenance during the planning period T. The model is also subjected to:

$$\sum_{t=\ell+1}^{\ell+T_i} x_{it} \geq 1 \quad \ell = 0, ..., T - T_i, \; i \in \mathcal{N}$$
$$x_{it} \leq z_t \quad t = 1, ..., T, \; i \in \mathcal{N}$$
$$x_{it}, z_t \in \{0, 1\} \quad t = 1, ..., T, \; i \in \mathcal{N}$$

2.2 Model 2

Parameters used in the model:

- $\mathcal{N}$ = [1,10], the set of components in the system.
- T = the number of time steps in the planning period.
- $T_i = \{42, 18, 90, 94, 49, 49, 34, 90, 37, 11\}$, the life of component $i \in \mathcal{N}$ measured in time steps.
- $c_{it} = [34\ 25\ 14\ 21\ 16\ 3\ 10\ 5\ 7\ 10]^T\,\mathbf{I}$, the cost of a new component $i \in \mathcal{N}$, $t \in 1, ..., T$, with $\mathbf{I}$ as the $1 \times T$. one-matrix.
- $d_t = 20\,\mathbf{I}$, the cost for maintenance occasion at time $t \in 1, ..., T$, with $\mathbf{I}$ as the $T \times 1$. one-matrix.

Variables used in the model:

- $x^i_{st} = \begin{cases} 1\text{if component i receives maintenance at time s and t but not in between} \\ 0 \text{ otherwise.} \end{cases}$
- $z_t = \begin{cases} 1 \text{ if maintenance is made at time } t \in 1, ..., T, \\ 0 \text{ otherwise.} \end{cases}$

The objective function we want to minimize this time is instead

$$min \quad z = \sum_{t=1}^{T} d_t z_t + \sum_{i=1}^{10} \sum_{(s,t)\in\Omega} c^i_{st} x^i_{st}. \tag{2}$$

This model is subjected to the following:

$$\sum_{s=0}^{t-1} x_{st}^{i} \leq z_t$$
$$\sum_{s=0}^{t-1} x_{st}^{i} = \sum_{r=t+1}^{T+1} x_{tr}^{i}$$
$$\sum_{t=1}^{T+1} x_{0t}^{i} = 1$$

3 Results

3.1 Question 1

When asked to solve the model given to us with $T = 125$ for the cases when both are required to be integers, x was relaxed with respect to the integrality and then when both variables are relaxed we got the values presented in Table 1.

State	Objective value	Time(s)
Integer requirement	762 €	5.3926
x relaxed	762 €	2.7623
Both relaxed	724 €	0.037035

Table 1: Objective value of our model as well as the computation time for the different cases using the large data file.

What we can see from the results is that relaxing only x generates no improvement in the objective value but it does improve the computation time. This is because when relaxing x, one could think of it as we now allow a component to be partly replaced. The problem is that we still have an integer requirement on z, which can be interpreted as the maintenance can only be done fully or not at all. This means that it does not matter that we allow the parts to be partly replaced since we can not partly replace it, which would result in calling for maintenance too often which ultimately leads to a worse objective value. The reason to why it is faster is because it is easier to optimize if we allow all sorts of values since it does not need to be checked by another constraint.

When relaxing both variables we can see an improvement in both objective value and computation time. Since we now allow both partial replacement as well as partial maintenance, the cost is reduced. The time for solving the problem was also improved, yet again because of a more free model.

The next question was to use the same model but with a different data file as well as with an added constraint, which is a valid inequality for small instances. The results of these computations can be found in Table 2.

State	Objective value	Time(s)
Integer requirement	14 €	0,0040016
Both relaxed	13.5 €	0,0009996
Both relaxed, added constraint	14 €	0.0020008

Table 2: Objective value of our model as well as the computation time for the different cases using the small data file.

One can see that the objective values as well as the computation time for the corresponding case is a lot smaller than the results in Table 1. This is due to the smaller data file, which only has two components and four time steps while the large data file has 10 components and 125 time steps. If we look at Table 2 we can see that the objective values are pretty similar and what separates them is the computation time. The reason to why they are so similar is because of the few time steps. Our components have a life length of three and four time steps respectively which means that if we only allow integers, there can only be a few scenarios that work if we want the system to fully function. Due to the few time steps, the computation time is also very fast.

When we relax both variables one can see that the objective value improves slightly since we can choose to partially fix parts of the system which makes sense because certain parts of the data file might not be optimal as a whole, but parts of them can create something better.

The added constraint is a valid inequality and does not cut away any feasible solutions. When we ran the program we saw that it had as many solutions with the added constraint as without.

3.2 Question 2

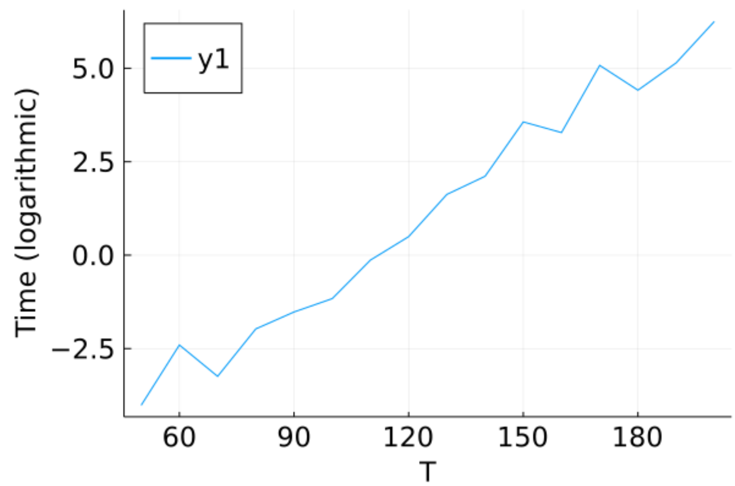

Figure 1: Figure showing the logarithmic computing time when T varies from 50 to 200. To get the "real" time, we need to take exp(logarithmic computing time).

From the figure above, figure 1, we can conclude that the time it takes to compute the optimal solutions increases exponentially when $T \rightarrow \infty$. The times varies from around 0.0183 seconds to almost 8 minutes and 38 seconds. This can be compared to a similar figure, figure 2, where the integrality constraints have been relaxed on both x_{it} and z_t. This results in a much faster computation time, as can be seen in figure 2.

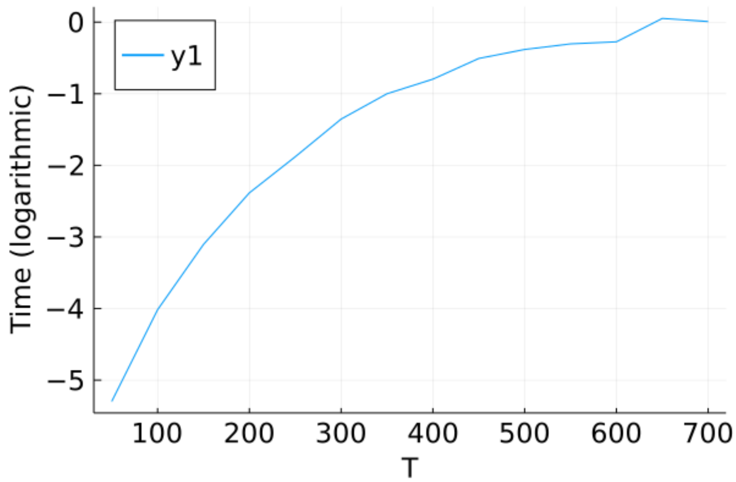

Figure 2: Figure showing the logarithmic computing time when T varies from 50 to 700. To get the "real" time, we need to take exp(logarithmic computing time).

In this figure, figure 2, we can see that the computing time varies from approximately 0.00525 seconds to 1 second. We can clearly see that when x_{it} and z_t has the integrality constraints the problem becomes more complex compared to without the integrality constraints.

When finding the optimal solution to the objective function, with Gurobi, it seams like the program needs more time to verifying the solutions optimality, then presolving or find a optimal solution, that is feasible. Example when $T = 150$ the presolving time is 0.5 seconds, while trying to find a solution took approximately 4 seconds and verifying that the solution is optimal took approximately 60 seconds.

3.3 Question 3

For the case when we relaxed the integrality requirement for both variables we saw a similar curve compared to Figure 2, which is displayed in Figure 4. The same similarity can not be seen in 1 and 3.

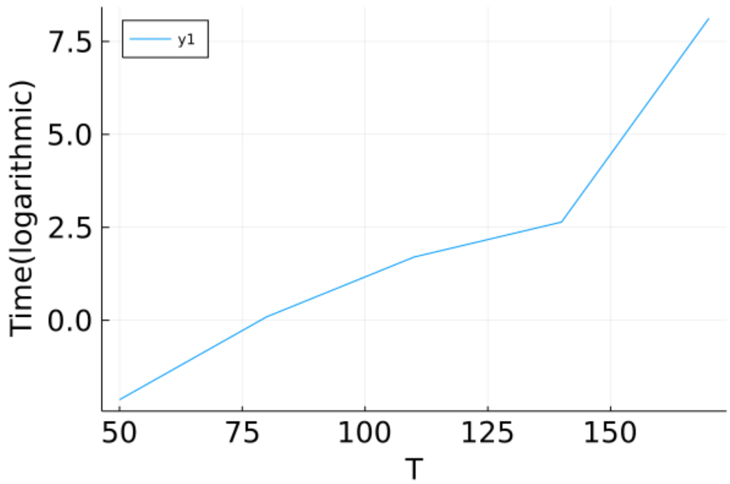

Figure 3: Figure showing the logarithmic computing time when T varies from 50 to 200 in the general model. To get the "real" time, we need to take exp(logarithmic computing time).

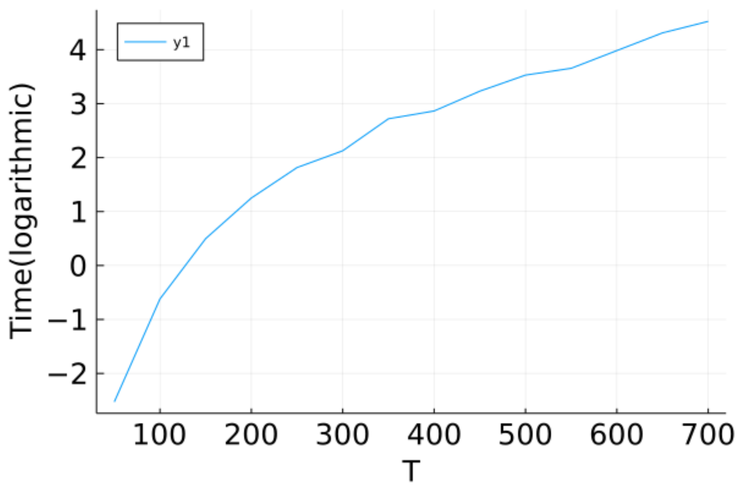

Figure 4: Figure showing the logarithmic computing time when T varies from 50 to 700 in the general model. To get the "real" time, we need to take exp(logarithmic computing time).

The key difference between figure 2 and 4, are the optimization time. That is since the general model is slower, which is to be expected since it has many more variables to take into account when optimizing. However, the general look of the graph is similar which means that when increasing the time period T the optimization time increases the same way in both models.

Why figure 1 and 3 are not similar can possibly be explained by the number of data points when T varies. In figure 1 we are using $T = \{50, 60, ..., 200\}$ however in figure 3 we used $T = \{50, 80, ..., 170\}$ since the optimization time where much larger, see earlier in the text for explanation. If we used more data points, i.e. $T = \{50, 60, ..., 200\}$ then figure 1 and 3 would probably be similar.

When looking at the two models, they have both similarities and differences. They are both 0-1 ILP models and with that follows several similarities, among them being that they both can be used as a maintenance scheduling models but also change quite a bit in the constraints. We are mainly thinking about allowing partial maintenance. Were it differs is how it calculates cost. In model 1, the cost for maintenance is measured at time T, but the general model treats the cost as it has happened over a period of maintenance intervals $t - s$. That is also the reason to why the objective value in the general model is slightly worse compared to the first model, since the general model has a constraint that ensures maintenance intervals at the endpoints. This will mean that if you find one point where you will have to make a maintenance, the first model will treat it as one cost whereas the general model will treat it as two maintenance intervals.

3.4 Question 4

If we add one constraint to the model 1, we can have the components survive to at least time period $T + r$, where $r \in \mathbb{N}$. See equation 3 for the new constraint.

$$\sum_{t=T-T_i+r}^{T} x_{it} \geq 1 \quad i \in \mathcal{N}. \tag{3}$$

In this constraint, constraint 3, we sum over the last part in the planing period for every component and set that we need to switch component sometime in that period to be able to survive to at least time period $T + r$. We can test that the solution fulfils the new requirement by giving two examples, one with $r = 10$ and the "normal" with $r = 0$.

Components	Survival time, r = 0	Survival time, r = 10
x_{st}^1	104	115
x_{st}^2	101	117
x_{st}^3	180	118
x_{st}^4	112	122
x_{st}^5	103	112
x_{st}^6	132	112
x_{st}^7	117	122
x_{st}^8	180	178
x_{st}^9	109	110
x_{st}^{10}	101	110

Table 3: Table showing the different lifespans of the ten components when they need to survive to at least time step $T + r$, $r = \{0, 10\}$, $T = 100$.

When $r = 0$ is the same values as the optimal solution without r and when $r = 10$ we can see, in table 4, that all the components survive till at least time step $T + r = T + 10 = 110$.

When choosing a value to r, we can only choose in $r = 1, ..., 11$, because $r = 0$ will have the "normal" optimal solution and $r > 0$. Furthermore, $r \leq 11$ since the shortest lifespan of a component is component 10 with a lifespan of 11 time steps. If we set $r > 11$ the component needs to survive to at least time step 112 and that is not possible when the last time step for maintenance is at time step= 100.

	r = 0	r = 1	r = 3	r = 6	r = 9	r = 10
Obj.Val.	615 €	615 €	635 €	645 €	660 €	670€
Comp.	Lifespan	Lifespan	Lifespan	Lifespan	Lifespan	Lifespan
x^1_{st}	104	107	120	112	111	115
x^2_{st}	101	101	104	106	109	117
x^3_{st}	180	137	176	168	159	118
x^4_{st}	112	130	186	148	185	122
x^5_{st}	103	114	127	127	122	112
x^6_{st}	132	121	109	127	118	112
x^7_{st}	117	106	126	122	125	122
x^8_{st}	180	162	141	108	127	178
x^9_{st}	109	102	105	107	110	110
x^{10}_{st}	101	101	103	106	110	110
Comp.	#MO.	#MO	#MO	#MO	#MO	#MO
x^1_{st}	2	2	2	2	2	2
x^2_{st}	5	5	5	5	6	6
x^3_{st}	1	1	1	1	1	1
x^4_{st}	1	1	1	1	1	1
x^5_{st}	2	2	2	2	2	2
x^6_{st}	2	2	2	2	2	2
x^7_{st}	3	3	3	3	3	3
x^8_{st}	1	1	1	1	1	1
x^9_{st}	2	2	2	2	2	2
x^{10}_{st}	10	10	10	11	10	11

Table 4: Table showing the lifespans, objective function and number of maintenance occasions (#MO) of the ten components when they need to survive to at least time step $T + r$, $r = \{0, 1, 3, 6, 9, 10\}$, $T = 100$.

If we study table 4, we can see some key differences in when r changes. We can see that the objective value changes from the "original" value of 615 to 670 when r changes, and that is because we need to replace some parts more times then we did before. Furthermore, the lifespan of the different components will also vary since r is changing. In addition we can compare the number of changed components with the "original" solution. There we can see that the second and tenth component is the only component that changes the amount of maintenance occasion on that component when r changes, i.e, when $r = \{6, 9, 10\}$. The difference between the "normal" solution and $r = \{6, 9, 10\}$ is only one or two extra maintenance occasion but will cost around 35 € to 55 € extra.