Finding a Lagrangean Lower Bound on the Emission Lot-Sizing Problem

Bart Mooij

Supervisor: Prof. dr. Albert P.M. Wagelmans
Second Reader: Mathijn J. Retel Helmrich MSc.

September 11, 2011

In this paper we have developed an algorithm that will find a Lagrangean lower-bound for the emission lot-sizing problem. Next to that we monitored the influence of different parameters on the quality of the lower bound and see how the associated computation time develops. This paper is part of the PhD. Research project of Mathijn J. Retel Helmrich on extensions of the classical economic lot-sizing problem.

Contents

1 Introduction

The lot-sizing problem, in its most basic form, is a relatively easy problem to solve. A feasible solution for the problem requires that a specific demand has to be satisfied in every period *1 ... T*. The decision has to be made in which period there will be production and how much to produce in those particular periods. If the decision to produce has been made, there are related set-up costs independent on the amount that will be produced. Next to those costs the actual production brings costs with it. If the production in a certain period exceeds the demand the remaining number of products will have to be stored. The storing of these products will also bring costs with it. For a producer it therefore is quite possibly not very convenient to produces the total demand for all the periods immediately in the first period, because this will bring great holding costs. On the other hand it is also quite possible that producing every period for that period alone is not so cost efficient either.

This basic problem has been solved for a few decades now but in the ever changing economy new problems arise constantly. Nowadays the entire world is concerned about the future of the planet. Thanks to the extensive use of fossil fuels, the world we live in is warming up. This increasing awareness of the CO_2 problem has consequences for every section of the economy. Everyone has to obey increasingly stricter rules regarding the CO_2-emission. The transport sector has to operate on cleaner vehicles, taxes on cars go up and the industry has to reduce its emission to levels set by their local government. Due to this extra restriction the optimal solution in the original problem will quit possibly no longer be a feasible solution.

The new problem still has to be solved again as cost efficiently as possible. This problem has been proven to be NP-hard and therefore an exact solution is not the goal in this paper. The goal is to gain a lower bound on the problem by means of a Lagrangean Heuristic. The next section will describe the problem we are dealing with in more detail. Section 3 and 4 respectively discuss the heuristic and the result. In section 5 we will draw the conclusions. This paper is a part of the Phd. research of M.J. Retel Helmrich MSc. into extensions of the classical economic lot-sizing problem.

2 Problem Description

So in this paper we want to obtain a solution for the extended lot-sizing problem. To gain such a solution a better understanding of the problem is useful. Therefore we start off by giving a detailed explanation of the problem. Like any other LP-problem the problem consists out of 4 parts respectively the decision variables, parameters, the objective function and the restrictions.

2.1 Standard

The problem can be divided into 2 major parts, where the first part is the standard lot-sizing and the second is the emission part. First the standard lot-sizing problem will be introduced and after that the additions, which follow from the emission part, will be specified.

2.1.1 Decision Variables

First up are the decision variables. Decision variables are the variables that are the most interesting, because of the fact that these variables are the ones that will lead to the actual solution. In this case we got 3 decision variables that are very closely related. We will define them as x_t, y_t and I_t. y_t indicates the periods in which production will find place. y is a binary variable meaning, $y_t \in \{0, 1\}$. This variable will be also be used to calculate the set-up costs, because every period where y_t equals 1 there is production and therefore also set-up costs. Apart from the decision in which period to produce we need also decide how much to produce in the periods where $y = 1$ and set-up costs are made. If the choice has been made to produce in a certain period t ($y_t = 1$) it is of course not efficient to produce nothing. We introduce the decision variable x_t which is $\in \mathbb{R}$, this variable indicates the amount that will be produces in the period$_\text{t}$. We could say that only this variable would be enough and that y is obsolete. In the next paragraph we will show why the introduction of y does have benefits. In some cases it is possible that the set-up costs are very high. When such a situation occurs you can intuitively imagine that producing for just 1 period ahead is not efficient. Namely this will result in a situation where the high set-up costs are made every period this is a situations that should be avoided. It might be more efficient to produce for more than 1 period. If in such a case the decision is made to produce more than the demand for that period t we will have some products left at the end of the period. Assuming that these products do not have an expiration date and we want to use them the next period we'll have to store them somewhere. The storage of these products will bring additional costs with them, which we will refer to as holding costs. The amount of units we want to take from period t to $t + 1$ is represented by I_t. As you can see it is more efficient to produce for multiple periods in the case where the set-up costs are very high. However this is not completely true, if in the same case the holding costs are also high it might become very inefficient to produce for a lot of periods. In a more extreme case where the set-up costs are low and the holding costs, you do not even want to produce for 1 period ahead. The lot-sizing problem is therefore a choice between making holding- or set-up costs. From this we can conclude that the decisions regarding the amounts to produce in the different periods are dependent on the ratio between the holding costs and the set-up costs. It doesn't matter whether the costs are high or low, but how high or low they are with respect to each other.

2.1.2 Parameters

In the previous section we mentioned some of the parameters already briefly. The formal formulations of the parameters, associated with this problem, are stated below. The first 4 parameters are the ones found in any lot-sizing problem. In section 2.2.1 the parameters, that are particularly for the problem where emission is to be considered, will be introduced.

Standard lot-sizing:

$$\begin{aligned} K_t &= \text{set up costs for period}_t && t = 1, \ldots, T \\ p_t &= \text{production costs for period}_t && t = 1, \ldots, T \\ h_t &= \text{holding costs for period}_t && t = 1, \ldots, T \\ D_t &= \text{demand in period}_t && t = 1, \ldots, T \end{aligned}$$

2.1.3 Objective Function

The purpose of the entire existence of the MIP-problems is to find a solution, which will results in a minimal, or sometimes a maximal, result. These minimizations are translated in the objective functions of the problems. In the lot-sizing problem the goal is to minimize the total costs, which consist of set-up costs, production costs and holding costs. The minimization is dependent on the relationship between the holding costs and set-up costs described earlier. The objective function for the standard lot-sizing problem becomes the following:

$$\min \sum_{t=1}^{T} (K_t y_t + p_t x_t + h_t I_t)$$

If we had only the objective function to deal with a computer program that solves the problems would choose our decision variable x as low as possible and even negative. To prevent this, the objective function has to be subjected to restrictions that prevent this from happening.

2.1.4 Restrictions

The restrictions are a very important part of every LP-problem for the reason previously described and the lot-sizing problem forms no exception. The standard lot-sizing problem has the following restrictions.

$$\begin{aligned} s.t. \qquad x_t &\leq D_{t,T} y_t && t = 1, \dots, T \\ I_t &= I_{t-1} + x_t - d_t && t = 1, \dots, T \\ x_t, I_t &\geq 0 && t = 1, \dots, T \\ y_t &\in \{0, 1\} && t = 1, \dots, T \\ I_0 &= 0 \end{aligned}$$

The 1^{st} restriction concerns the maximum production for every period t. We want no inventory left at the end of period T and therefore we set a maximum to the production in period t, the periods maximum is equal to the demand of all the remaining periods. In that way there cannot be produced more than will be demanded. The 2^{nd} restriction deals with the inventory I_t for every period t. It states that the inventory in a period t has to be equal to the inventory from the previous period I_{t-1} plus the production in period t x_t minus the demand D_t. The next restriction ensures that we don't have a negative produciton or inventory. The y_t ensures that there can only be production if set-up costs are made in that period. This is where the y_t decision variable makes itself useful for the first time. The third restriction makes sure that negative production or inventory is not possible. Without these restrictions a solver can for example decide to produce negative to obtain a lower total cost. The last restrictions respectively state that we start out without any inventory and the $y \in \mathbb{B}$.

2.2 Emission

The second part of the problem we are dealing with in this paper, are the additions made to create the actual emission lot-sizing problem. The difference lies in the extra emission restriction and its corresponding parameters.

2.2.1 Additional Parameters

Before we introduce the import part, the restriction, we introduce the different new parameters.

$$\begin{aligned}
\hat{K}_t &= \text{set up emission costs for period}_t && t = 1, \ldots, T \\
\hat{p}_t &= \text{production emission costs for period}_t && t = 1, \ldots, T \\
\hat{h}_t &= \text{holding emission costs for period}_t && t = 1, \ldots, T \\
\hat{C} &= \text{emission capacity}
\end{aligned}$$

The parameters stated above can be seen as the CO_2 costs for their corresponding actions. Every company or organization has to keep to sum of the emission of all its processes below some level. This will lead us to the following constraint which this paper is actually all about.

2.2.2 Additional restriction

Without this additional restriction there wouldn't be a problem so this restriction is the most important part of the problem.

$$\sum_{t=1}^{T} (\hat{K}_t y_t + \hat{p}_t x_t + \hat{h}_t I_t) \ \leq \hat{C}$$

It's relatively straightforward but like said before without it the lot-sizing problem is exactly solvable in polynomial time.

2.3 Data Sets

The data we used to test our methods are provided by Retel Helmrich, M.J. and are introduced in the this section Every problem instance consists out of 8 separate files, each of which follows from a specified distribution. Every problem instance has different properties according to the parameters used for the generation process. Of every combination of parameters 10 problems are generated.

- **A**: Indicates the properties of the dataset 0 if the dataset posseses the zero inventory property, 1 if the dataset does not have this property and 4 if the dataset has cheap & dirty periods alternating with clean & expensive periods.
- **K**: Indicates the mean of the uniform distribution from which the setup costs are generated.

$K_t \sim$ discrete uniform[500,1500], discrete uniform[2500,7500], discrete uniform[5000,15000].

- $\hat{K}$: Indicates the mean of the uniform distribution from which the set-up emission costs are generated.
  $\hat{K}_t \sim$ discrete uniform[500,1500], discrete uniform[2500,7500], discrete uniform[5000,15000].
- **T**: T $\in [25, 50, 100]$ indicates the amount of periods in that problem instance.

The demand, the (emission)production, costs are from the same distributions for all of the problem instances and they are generated as follows:

- **d**: $d_t \sim$ discrete uniform[0,200]
- $p_t, \hat{p}_t, h_t, \hat{h}_t \sim$ discrete uniform[0,20].

According to this division we can later on say something about the development of the calculation time and the quality of the lower bound for the different problem instances to see which parts have the most influence. The optimal value for each of the problem instances are also provided, these are obtained using CPLEX 10.1.

3 Methods

Like described in the previous sections the problem with the emission constraint is NP-hard and can probabaly not be solved exactly in polynomial time. Therefore an alternative method has to be constructed to solve it. The main goal in this paper was to find a lower bound. Using different kind of methods a heuristic will be developed to do so.

3.1 Wagner Within

One of the methods we will be using in our algorithm is the Wagner-Whitin algorithm. The standard lot-sizing problem can be solved using this. It shows how we can solve the problem by looking at it as a shortest path problem. In the paper *'An Efficient Implementation of the Wagner-Whitin Algorithm for Dynamic Lot-sizing'* [1] they present an efficient computer implementation of the algorithm. However because of the emission restrictions this algorithm cannot be applied immediately first the problem has to be modified.

3.2 Lagrange Relaxation

In general a good way to find the lower bound on a problem is the Lagrange relaxation method. The Lagrange method relaxes a restriction to simplify the problem and this is exactly what we need to do before we can use the Wagner Whitin algorithm. The Lagrange method works as follows. For a certain LP-problem we have the following equations

$$\begin{aligned} &min\ c^T x \\ &s.t. \qquad Ax\ \leq b \end{aligned}$$

Next we divide up the restrictions in 2 parts one of which we later will relax into the objective function.

$$\begin{aligned} &min\ c^T x \\ &s.t. \qquad A_1 x \qquad \leq b1 \\ &\qquad\qquad A_2 x \qquad \leq b2 \end{aligned}$$

When we relax the A_2 part into the objective function we get the following problem.

$$\begin{aligned} &min\ \ c^T x - \lambda^T (b_2 - A_2 x) \\ &s.t. \qquad A_1 x\ \leq b_1 \\ &\qquad\quad \lambda \quad\ \geq 0 \end{aligned}$$

The λ works as a penalty. When the restriction imposed in A_2 is violated it means our objective function will be increased, which is something that must be avoided. Any solver will therefore try to 'minimize' the penalty and in that way get a solution that violates the relaxed restriction as little as possible. The benefit of the Lagrange relaxation method is that we can manipulate our problem. If we apply this method to our problem we can change it in such a way that we can again use the Wagner Within method described in the previous section. If we relax the additional constraint regarding the emission the following problem will emerge.

$$\begin{aligned} \min \qquad & \sum_{t=1}^{T} (K_t y_t + p_t x_t + h_t I_t) + \lambda \sum_{t=1}^{T} (\hat{K}_t y_t + \hat{p}_t x_t + \hat{h}_t I_t - \hat{C}) \\ &= \sum_{t=1}^{T} ((K_t + \lambda \hat{K}_t) y_t + (p_t + \lambda \hat{p}_t) x_t + (h_t + \lambda \hat{h}_t) I_t - \lambda \hat{C}) \end{aligned}$$

$$\begin{aligned}
s.t. \qquad x_t &\leq D_{t,T} y_t && t = 1, \ldots, T \\
I_t &= I_{t-1} + x_t - d_t && t = 1, \ldots, T \\
x_t, I_t &\geq 0 && t = 1, \ldots, T \\
y_t &\in \{0, 1\} && t = 1, \ldots, T \\
I_0 &= 0
\end{aligned}$$

If we now take a fixed number for λ we will again have standard lot-sizing. Only with an objective function which has costs, for holding production and set-up that also contain the emission costs and are at least as high, depending on the λ. Therefore applying the Lagrangean relaxation will be the first step into solving our problem.

3.3 Algorithm

In this section we are going to explain the algorithm that will provide us with a lower bound using the techniques described earlier. In the first part an example is given and in the second part the formal algorithm is presented in the form of a pseudo code. First we will find 2 initial solutions using λ_{min} and λ_{max} in the dual problem. These solutions will be obtained using the Wagner Within algorithm, which is possible because in these cases we know what λ is. From this point on the a method from *”Eisner and Severance”* [2] will be used along with the Wagner Within method. This method is described in detail in *Sensitivity Analysis in Combinatorial”* the PhD research from A.P.M. Wagelmans [3] and is shown in the next few figures.

From the initial solutions x we can now define 2 linear functions in the terms of lambda. These 2 linear equations are shown in figure 1. The ascending is the one where $\lambda = 0$ and the descending one represent the solutions gained with λ_{max}.

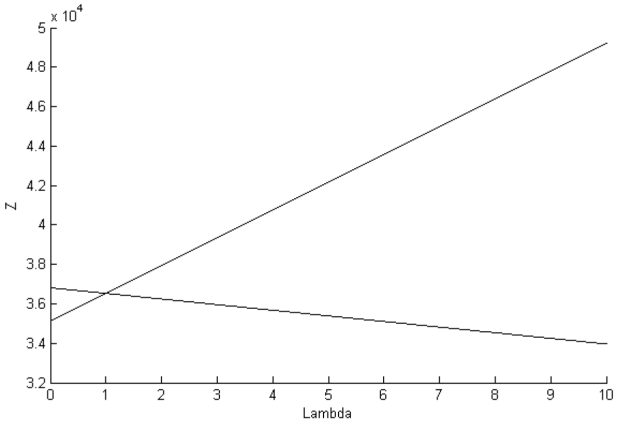


Fig. 1: step 1

In figure 1 we can see the 2 functions that are obtained from using λ_{max} and λ_{min}. The solution that is with λ_{max} has a negative gradient and the one derived from λ_{min} a positive one.

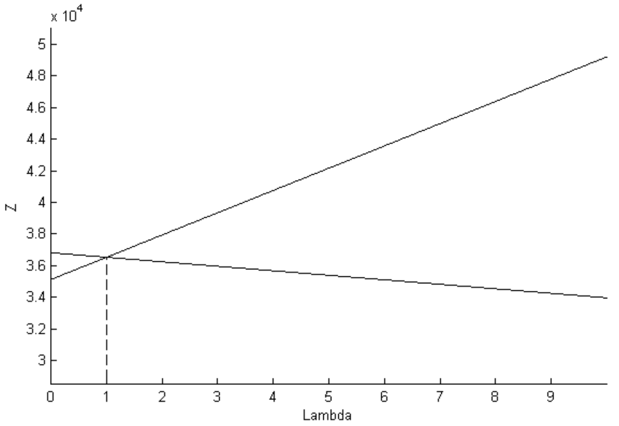


Fig. 2: step 2

Figure 2 shows the $\bar{\lambda}$ for which the 2 equations intersect. We use this λ in the way we earlier used the λ_{min} & λ_{max}. If a new solution is found there will also be a new equation with it. This new equation is shown in the next figure.

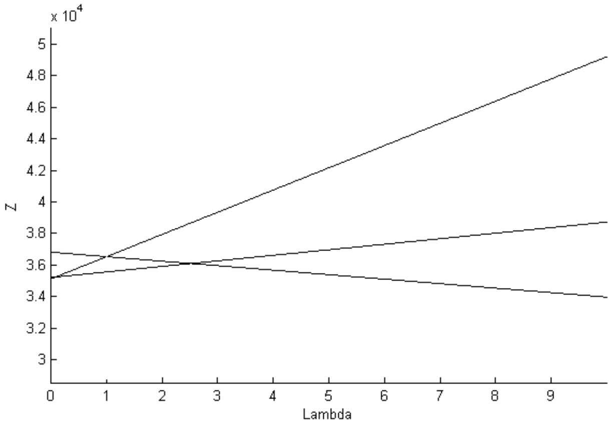


Fig. 3: step 3

In figure 3 we can see that for $\bar{\lambda}$ the new equation has a lower z-value. If this wouldn't be the case the new equations wouldn't have much effect on the lower envelope and would be discarded and the algorithm is done. In this case however the value is lower and we keep starting the procedure over again with 1 of the 2 new λ values where the lines intersect. When the collection of breakpoints is depleted the graph can come to look something like figure 4. The maximum of the lower envelope will be the estimation of our lower bound.

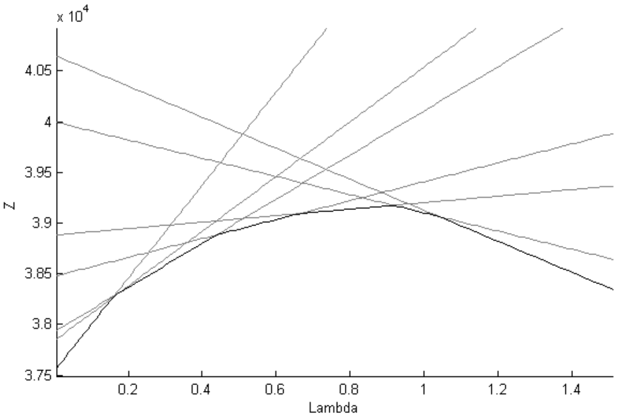
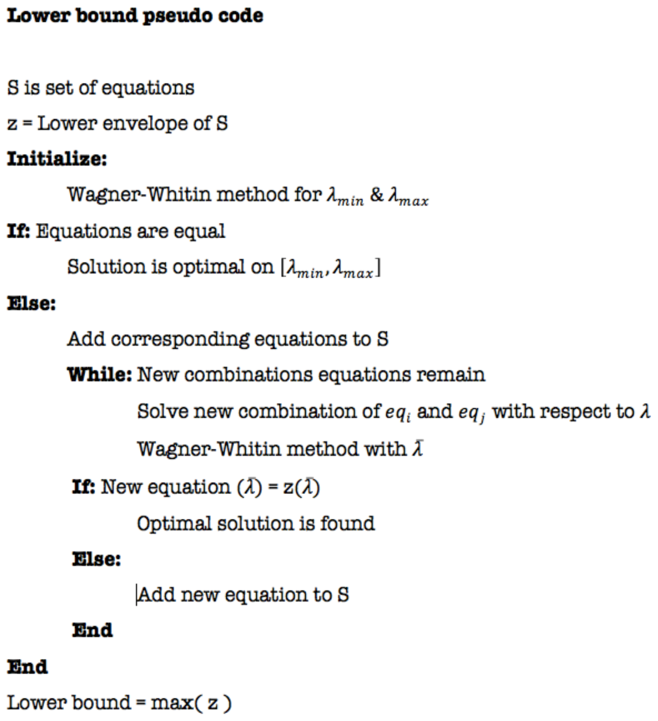


Fig. 4: final

3.3.1 Pseudo code

The intuitive way presented in the preceding section will now be translated into a pseudo code so it can be implemented in a computer program, in this paper we make use of MATLAB 2010.



```
Lower bound pseudo code

S is set of equations
z = Lower envelope of S
Initialize:
    Wagner-Whitin method for $\lambda_{min}$ & $\lambda_{max}$
If: Equations are equal
    Solution is optimal on $[\lambda_{min}, \lambda_{max}]$
Else:
    Add corresponding equations to S
    While: New combinations equations remain
        Solve new combination of $eq_i$ and $eq_j$ with respect to $\lambda$
        Wagner-Whitin method with $\bar{\lambda}$
    If: New equation ($\bar{\lambda}$) = z($\bar{\lambda}$)
        Optimal solution is found
    Else:
        Add new equation to S
    End
End
Lower bound = max( z )
```

Fig. 5: The pseudo code of the algorithm that finds a lower bound on the emission lot-sizing problem

4 Results

After implementing the method we previously described we want to know how well it performs. In this section we are going to look at the performance in two ways. First we want to know how close we get to finding an optimal value. However this isn't the only important aspect of a solution, we also want to know how long a computer will be needing to come to that solution. A method, which leads to a value less than 1% away from the optimal, with a computation time of 1 second, will sometimes be considered better than a method that leads to the optimal value with 2 seconds of computation time.

4.1 Optimal Values

The solution found by the heuristic will be evaluated by comparing it with the optimal value of the problem instance. Keeping in mind we are searching for a lower bound the desired results is that we find only negative numbers and as close to zero as possible. The results found by the heuristic are presented in the table below and are ordered by the different problem instances. In most of the analysis the $A = 4$ problems have been left out of the analysis, because only for $K = 1000$ the optimal values were available.

			A								
			0			1			4		
		T \ K̂	1000	5000	10000	1000	5000	10000	1000	5000	10000
K	1000	25	-0.4317	-0.5947	-0.6248	-0.9387	-0.9634	-0.8881	-	-3.0652	-3.8810
		50	-0.2303	-0.2568	-0.2114	-0.3730	-0.3559	-0.3510	-	-1.2897	-2.0955
		100	-0.0718	-0.0651	-0.0806	-0.1213	-0.1228	-0.1196	-	-0.5591	-0.7863
	5000	25	-1.6059	-1.1062	-1.5026	-1.4442	-1.9858	-1.6688	-	-	-
		50	-0.4565	-0.5264	-0.5478	-0.5778	-0.6097	-0.6784	-	-	-
		100	-0.1644	-0.1940	-0.1828	-0.2806	-0.2229	-0.2303	-	-	-
	10000	25	-0.9533	-2.4631	-1.2127	-2.4496	-2.0941	-1.8019	-	-	-
		50	-0.5694	-0.6300	-0.6308	-0.6839	-0.8366	-1.0270	-	-	-
		100	-0.2445	-0.2470	-0.2055	-0.2233	-0.2985	-0.2582	-	-	-

Tab. 1: Shows the mean deviation from the optimal value in percentages for each of the 10 problem instances associated with the A,K,K̂,T combination. Not every optimal solution was provided and therefore not every calculation could me made

4.1.1 Parameter Influence

The results we found can are divided in different section according to the parameters of the dataset by which they are computed. In the table we can distinguish the 4 different parameters and from table 1 we can see how they influence the quality of the lower bounds.

A Because of the fact the the algorithm is based on the zero-inventory property we expect it to work better on the problem instances where the zero-inventory property holds ($A = 0$). By comparing the different columns of A we can see that our expectations turned out to be correct. Only in 3 of the 27 problems the opposite is true. If we compare $A = 0$ with $A = 1$ we can see that on average the lower bound with $A = 0$ lies 0.21 percent closer to the optimal value. If we do the same for the available sections from $A = 4$ with the same section of $A = 1$ we see that the difference is here 1.29 percent on average.

K The parameter K represents the set-up costs and shows that the quality of the lower bound decreases when the parameter increases.

$\hat{\text{K}}$ The set-up emission costs do not show a clear pattern in terms of the lower bound quality. In some cases the quality decreases when the parameter increase, such is the case where $A = 0$, $K = 1000$ and $T = 25$. In other cases however the quality goes the opposite way for example where $A = 1$, $K = 10000$ and $T = 100$. In it appears even so that in some cases no pattern emerges at all, the quality increases and decreases if we only change $\hat{\text{K}}$. Therefore we can conclude that $\hat{\text{K}}$ has probably very little to no effect.

T Where the set-up emission costs had no clear pattern at all, the amount of periods, T, has a very clear pattern. We can see that when the problem contains more periods the quality actually increases. This increase is independent of all the other parameter, in each possible combination the same pattern emerges. We take another look at this specific parameter later on.

4.1.2 Optimal Value Statistics

To get a better view of 1 of the most important parameters another division has been made in table 2. In table 1 we already saw that there was a positive relationship between the number of periods and the quality of the lower bound. In table2 we take a closer look at this relationship. We can again see clearly that there is a positive relationship. In problems where 100

	T		
	25	50	100
Mean	-1.4045	-0.4990	-0.1843
Std	1.4707	0.6550	0.2341
Std as % of Mean	104.7115	131.2557	126.0160
Max	9.2654	4.3154	1.6707

Tab. 2: Shows the Mean, Std and Max.

periods are being considered the algorithm is performing better in terms of the mean. The standard deviation appears on first sight also to get better, here we assume lower is better, but if we take a look at line number 3 'Std as % of Mean' we see that the standardized standard deviations get worse. This means on average we get a better performance but the variation gets larger and therefor it is more difficult to make prediction if we would want to.

4.2 Computation Time

An important aspect of developing and implementing a heuristic is the computation time it will take to find results. It was therefore monitored how long it took to solve each of the problem instances mentioned in the dataset section. The solving of the problem instances was performed using a MacBook with a 2.26 Ghz Intel Core 2 Duo processor.

			A								
			0			1			4		
		$\hat{\text{K}}$ T	1000	5000	10000	1000	5000	10000	1000	5000	10000
K	1000	25	1.1030	1.2201	1.1973	2.1145	2.0420	2.0294	1.6855	2.0232	2.3736
		50	2.4793	3.2178	2.9766	4.2785	5.1353	4.5190	5.8275	5.4266	6.1770
		100	7.6717	7.5649	8.0108	13.8581	14.0068	12.9579	19.5100	20.5312	23.3218
	5000	25	0.9094	0.9802	1.0418	1.1624	1.2146	1.3493	0.7541	0.7625	0.0839
		50	2.2631	2.1088	1.9393	2.8703	2.5966	2.4644	2.2366	1.8696	2.3905
		100	4.8501	5.7861	4.9514	6.7289	6.8143	6.9660	6.7805	6.4062	7.9065
	10000	25	0.8546	0.6384	0.9026	0.9118	0.9522	0.9858	1.0677	0.5217	0.8140
		50	1.8987	1.4589	1.6610	1.9005	1.7944	1.9924	3.3974	1.3791	1.3676
		100	4.3827	4.0902	4.9900	5.1302	5.0641	5.4425	10.9046	4.3577	3.8861

Tab. 3: Shows the time it takes on average for each of the 10 problem instances associated with the A,K,$\hat{\text{K}}$,T combination.

4.2.1 Parameter Influence

The results we found can again be divided in different section according to the parameters of the dataset by which they are computed. In the table we can distinguish the 4 different parameters and at the hand of those we can see what their influence is on the computation time.

A The first distinction that is interesting is the distinction between $A = 0$ and $A = 1$. Up front we would have expected that the computation time for $A = 0$ will be shorter due to the zero inventory property. Looking at the table we can see that this in fact is the case for all of the problem instances. The other problem which is identified by $A = 4$ shows no clear pattern if compared to the other possibilities of A. Sometimes it performs better in terms of time, but often it takes longer to solve the problems.

T Another important feature of the problem instances is the number of periods involved 25, 50 or 100 respectively. In this case it speaks for itself that if the planning horizon increases, the calculation time increases with it. This expectation is confirmed by the results for all the combinations of the other parameters we can see that the time is always ascending. The computation time multiplies with 2.85 on average if the periods doubles from 25 to 50. When we apply the same calculation to the increase from 50 to 100 periods the computations times multiplies with 3.50 on average.

//

K̂ If we take a look at the Parameter $\hat{K}$ we can see that there for an changing $\hat{K}$ there isn't a clear pattern with respect to the computation time, in some cases it causes more computation time and in some cases less.

K The parameter K shows a more interesting development. One might expect the computation time to get larger when the parameter increases, but exactly the opposite is true. When we take a look at $A = 0$ we see that the computation time for the problem with $K = 1000, \hat{K} = 10000$ and $T = 100$ the computation time is around 8 seconds where the same problem only with $K = 10000$ only takes 5 seconds. From the dataset section we know that K represent the set-up costs. Knowing this we can actually explain the results found. When the set-up costs are large, it isn't efficient to produce in many periods and therefore fewer solutions will be considered. From $A = 4$ we know that there is 1 production period more where T was 25, but since the differences aren't that big we can probably neglect it. When we take a look at the differences with respect to T we see that the decrease is even larger. In this part the 'cheap and dirty vs. clean and expense' data sets are considered. They appear to have even fewer solutions that are considered and therefore it becomes even quicker.

4.2.2 Calculation Time Statistics

To again get a better view of 1 of the most important parameters another division has been made in table 4. The table shows the mean of the problems, which reacts as expected to the different problems. The $T = 25$ problems have a shorter calculation time than those from $T = 50$. which on their turn are shorter than $T = 100$. The same development occurs with the standard deviation. In general problems that deal with larger number will have larger standard deviations most of the time. Therefore it is difficult to compare the standard deviation of problems with different number of periods. To compare the standard deviations anyway we have taken the standard deviation as percentage of the mean. In this way we can compare different periods with each other. If we compare the different periods after this modification we can still see the same development we saw earlier.

	T		
	25	50	100
Mean	0.8364	2.4792	8.0371
Std	0.4750	1.6685	5.8414
Std as % of Mean	56.7914	67.3001	72.6803
Min	0.1702	0.3830	1.9685
Max	2.6721	12.4629	29.1217

Tab. 4: Shows the Mean, Std, Std as percentage from Mean, Min and Max

5 Conclusions

There are 2 parts we have been monitoring in this paper. We check the quality of the lower bounds achieved through our algorithm and the computation time needed to find the results. We now sum up the most important conclusions of our findings.

5.1 Quality of the Lower Bounds

The algorithm delivers the expected results in terms of the type of problem instances. The zero-property datasets perform the best followed by the one without this property and finally followed by the instances where the cheap and dirty vs. clean and expensive property is present. The previous result was to be expected up front. The next one however was unexpected. One of the most interesting findings is the one about the relationship between the number of periods and the quality of the lower bound. It shows that the algorithm performs better on average with the problems with a larger amount of periods.

5.2 Computation Time

There are a few conclusions we can draw about the influence of the different parameters on the computation time. First up are the different problems we have been looking at in this paper. Problems that possessed the zero-inventory property have shown to take less long than problems that do not posses that property. The problems which are dealing with the 'cheap and dirty' vs. 'expensive and clean' have shown to be difficult to say something about. Next is the one that is the most logical, the T that represents the number of periods. The larger the amount of periods that is being considered the larger the computation time will be, assuming the rest of the problems remains the same. This is very intuitive and was to be expected. From table 4 we can see that the variation in calculation time increase with T. A parameter where no distinct pattern has been proven is the set-up emission costs. The results vary too much to say something about it. The normal set-up costs however do seem to influence the computation time. The higher the computation time becomes the shorter it takes to solve the problem.

6 References

1. James R. Evans., ”An Efficient Implementation of the Wagner-Whitin Algorithm for Dynamic Lot-Sizing” *Journal of Operations Management*, published 2 Feb. 1985; Vol. 5 No. 2 pp. 229-235.

2. Eisner, M.J. and and Severance, D.G. (1976), *”Mathematical techniques for efficient record segmentation in large shared databases”*, Journal of the Association for Computing Machinery (1976) 619-635

3. Wagelmans, A.P.M (1990). *”Sensitivity Analysis in Combinatorial Optimization”*. PhD. Thesis. Erasmus University: The Netherlands.