BSc Thesis

Performance of on-line lot sizing heuristics with a worst-case ratio of two

Florian Maas*

Supervisor: W. van den Heuvel

Erasmus School of Economics

July 2013

Abstract

Research has shown that for the economic lot-sizing problem, the on-line heuristics PPA and H* are amongst the best performing heuristics on the test bed by Berry (1972). Both of these heuristics have a worst-case ratio of 2, which is the lowest value possible for heuristics in the class of on-line heuristics. This might raise the presumption that the possession of a worst-case ratio of 2 is a good property for an on-line heuristic to have. We shall do this by modifying some well-known existing heuristics in a way that gives them a worst-case ratio of 2, and then comparing their performance against the performance of the original heuristics using the test bed by Simpson (2009).

*Studentnumber: 342750 E-mail: fpgmaas@gmail.com

Contents

1 Introduction

The Economic Order Quantity (EOQ) model deals with determining the size and frequency of replenishments which minimize the inventory-related costs, such as the order costs and the holding costs. In the EOQ model, there is a fixed order cost K incurred for every order placed, and a holding cost h associated with holding one item in inventory for one unit of time. Furthermore, there is a deterministic demand rate D, which is assumed to be constant over time. With this information one can calculate the optimal order quantity:

$$Q* = \sqrt{\frac{2KD}{h}}$$

A well-known property of the EOQ is that the order costs and the holding costs are perfectly balanced in the optimal solution (Silver et al., 1998).
However, the assumption of the demand being constant over time does certainly not hold in all inventory-related environments. Relaxing this assumption allows us to deal with a much greater variety of practical situations. The problem which deals with minimizing the inventory-related costs with a demand rate that is allowed to vary over time is called the Economic Lot-Sizing (ELS) problem. Since the demand is now no longer equal in every period, we can no longer assume that it is optimal to always use the same replenishment quantity or interval, as was the case with the EOQ model. A method for finding the optimal solution to this problem was given by Wagner and Within (1958), however this approach has a few drawbacks (Silver et al., 1998):

- The relatively complex nature of the algorithm makes it more difficult for the practitioner to understand than other approaches.
- There is a possible need for a well-defined ending point for the demand pattern.
- When one operates on a rolling schedule, replenishment quantities from the past should not be altered when new information becomes available. This is called the *insulation* property by Baker (1989). Unfortunately, the Wagner-Within algorithm does not have this property.
- While heuristics can usually be easily modified to allow for continuous opportunities to replenish, the computational complexity of the Wagner-Within algorithm goes up rapidly when modified for this purpose.

For this reasons, a large quantity of heuristics has been developed ever since. In this thesis we will consider the class of heuristics as described in Van den Heuvel and Wagelmans (2010), the on-line lot sizing heuristics. The on-line lot sizing heuristics are defined as follows:

Definition On-line lot-sizing heuristics make setup decisions period by period (so previously made decisions are fixed and cannot be changed) and setup decisions do not depend on future demand.

Clearly, all heuristics in this class satisfy the earlier mentioned desirable insulation property. Before announcing exactly which heuristics will be considered in this paper, it is important to notice that there is a lot of ambiguity in the naming of algorithms in the literature on this topic. For example, Silver et al. (1998) use the PPA results of Baker (1989) as PPB results in table 6.10, while they should have used the results from LTC. For the sake of clarity we shall use the same names for the algorithms as Baker (1989) whenever possible.

Heuristic	H*	LPC	LTC	LUC	PPA
Worst-case ratio	2	∞	3	∞	2
Average percentual deviation	0.664	0.943	10.274	9.303	1.339

Table 1: Heuristics with their worst-case ratios and their performance on the test bed by Berry (1972)

There are two ways of assessing the performance of heuristics. The first method consists of empirical research, which is mainly done by running the algorithms on a (simulated) data set, see for example Simpson (2009) and Baker (1989). The second method consists of performing analytical research, which can be divided into probabilistic and worst-case analysis, see Axsäter (1982), Van den Heuvel and Wagelmans (2010), and Bitran et al. (1984). Probabilistic analysis mainly focuses on the average performance of heuristics, while worst-case analysis aims to find upper bounds on the worst-case performance of the heuristics.
Amongst others, Van den Heuvel and Wagelmans (2010) and Axsäter (1982) have performed worst-case analysis on various heuristics. Some heuristics along with their worst-case ratios and their performance on the test bed by Berry (1972) are shown in Table 1 . It is notable that the heuristics which have a worst-case ratio of exactly two are amongst the best performing heuristics in this test bed, while the performance of the heuristics which have a worst-case ratio which is strictly higher than two ranges from very bad to very good. Of course no conclusions can be drawn from this, but this does raise the presumption that the possession of a low worst-case ratio is a good property for a heuristic to have. Since Van den Heuvel and Wagelmans (2010) have already shown that all the heuristics in the class of on-line heuristics have a worst-case ratio of at least two, this is the same as stating that having a worst-case ratio of two is a good property for an on-line heuristic to have. In this thesis, we shall test whether this is indeed the case. In Section 2 we shall explain and illustrate the various algorithms and heuristics, and modify some existing heuristics on which we shall base our research. We shall than evaluate the performance of these modified heuristics and compare it with the performance of the existing heuristics in Section 3, in which we shall also evaluate the influence of certain factors of the test bed on the performance of the heuristics. In Section 4 we shall draw our conclusions based on the results of the foregoing analysis.

2 Algorithms and Heuristics

We shall now present the various implementations of the Wagner-Within algorithms, along with descriptions of the heuristics used. Since we will use Simpson (2009) to evaluate the performance of the heuristics in Section 3, we shall use the algorithms in a dynamic fashion. When one uses the algorithms statically, all information up until the end of the planning horizon is known, and all this information can be used for placing the first order. However, this is not very realistic in practice, since one often does not know the demand for a period which lies far in the future, or the demand for that period is at least a lot more uncertain. So in the dynamic problem, the algorithms only have information for a certain horizon length which they can use for planning the first order. When this order is placed, the horizon is rolled forward to the first period for which the demand is not yet satisfied, and now the heuristic can use the information of the horizon from this point for planning the next order. This is done until demand for all periods is satisfied.

All solution approaches are based on the same set of assumptions, as described by Silver et al.

(1998). The assumptions are as follows:

- The demand rate is given in the form of d_j to be satisfied in period j ($j = 1, 2, ..., N$) where the planning horizon is at the end of period N. Of course, the demand rate may vary from one period to the next, but it is assumed known.
- The entire requirements of each period must be available at the beginning of that period. Therefore, a replenishment arriving part-way through a period cannot be used to satisfy that period's requirements; it is cheaper, in terms of reduced carrying costs, to delay its arrival until the start of the next period. Thus, replenishments are constrained to arrive at the beginnings of periods.
- The unit variable cost does not depend on the replenishment quantity; in particular, there are no discounts in either the unit purchase cost or the unit transportation cost.
- The cost factors do not change appreciably with time; in particular, inflation is at a negligibly low level.
- The item is treated entirely independently of other items. That is, benefits from joint review or replenishment do not exist or are ignored.
- The replenishment lead time is known with certainty (a special case being zero duration) so that delivery can be timed to occur right at the beginning of a period.
- No shortages are allowed.
- The entire order quantity is delivered at the same time.
- For simplicity it is assumed that the carrying cost is only applicable to inventory that is carried over from one period to the next. It should be emphasized that all three approaches can easily handle the situation where carrying charges are included on the material during the period in which it is used to satisfy the demand requirements but, for practical purposes, this is an unnecessary complication.

To clarify the algorithms, we shall use the test case as presented in Table 2, where the holding cost (h) is \$1/unit/month, and the fixed order cost (K) is equal to \$320 for every order placed.

Period	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100

Table 2: The test case

2.1 Wagner-Within

In this section three versions of the Wagner-Within algorithm shall be presented. First, the standard Wagner-Within algorithm shall be elaborated, which treats the entire data set as one static problem. Of course, no single heuristic can recreate this optimal solution under a constrained planning horizon if some orders are further apart than the planning horizon in the optimal solution. Therefore, the constrained Wagner-Within algorithm is introduced. Also, the Wagner-Within algorithm can be used as a heuristic by using the static Wagner-Within algorithm in a dynamic environment.

2.1.1 Static Wagner-Within algorithm

The Wagner-Within algorithm was developed by Wagner and Within (1958). The functioning of this dynamic algorithm shall be briefly described here. Suppose we know d_j for periods $1, ..., N$, and let $F(t)$ denote the total costs belonging to the best replenishment strategy for the periods $1, ..., t$. We shall use the test case from Table 2 to illustrate how the algorithm continues. Clearly, $F(1)$ is equal to the order cost, since the only and thus the best way of satisfying the demand for period 1 is simply to order the quantity demanded. However, for determining $F(2)$, we now have two options. We can either cover from the beginning to the end of period 1 in the best way we can, and than place an order at the start of period 2 to fulfill its' demand (option1), or we can order the quantity demanded for both periods 1 and 2 at once and carry the required 100 items for period 2 from the first period (option 2). The associated costs are:

$$\begin{aligned}
\text{Costs of option 1} &= F(1) + K = F(1) + \$320 \\
&= \$320 + \$320 \\
&= \$640 \\
\text{Costs of option 2} &= K + h \times d_2 \\
&= \$320 + \$1 \times \$100 \\
&= \$420
\end{aligned}$$

We see that the best way to cover the demand of the first two periods is option 1, and therefore $F(2) = \$420$

Now for determining $F(3)$, we have a total of three options. We can use the best strategy to fulfill the demand for the first two periods, and place a new order for the requirements of period 3 (option 1). Another option is to fulfill the demand for the first period in the best way we can, and place one order for the demand requirements of periods 2 and 3 at the beginning of period 2 (option 2). The third option is to place one order at the beginning of period 1 to fulfill the demand requirements for periods 1 to 3 (option 3). Again, one can calculate the corresponding costs and set $F(3)$ to be the minimum of these numbers. One can determine the optimal costs by continuing in this fashion until $F(N)$ is reached.

A useful representation for implementing this method is given by (Simpson, 2009), which is as follows: Define Z_{kt} as the total costs of placing an order in period k that replenishes periods k through t, which can be calculated by the following expression:

$$Z_{kt} = K + \sum_{j=k+1}^{t} (j-k)hd_j$$

One can then determine the costs corresponding of satisfying the demand up until period t as follows:

$$F(t) = \min\{Z_{kt} + F(k-1) | k = 1, ..., t\}$$

The optimal schedule can be found through backward induction. Start by finding $\min_k(Z_{kN})$. The k for which this expression was minimal is the time in which the order should be made fulfilling the demand for periods k through N. Now set $k^* = k - 1$, and find $\min_k(Z_{kk^*})$. Now, the k for which the minimum has been attained is the period in which an order should be placed fulfilling the demand for periods k through k^*. Continuing in this fashion until the first period is reached results in the optimal replenishment schedule.

	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100
Replenishments	180	0	325	0	0	0	225	0	375	0	0	0
Ending inventory	100	0	200	100	50	0	125	0	250	150	0	100
											Total costs:	$2355

Table 3: Results of applying the Wagner-Within algorithm on the test case

2.1.2 Constrained Wagner-Within

The static Wagner-Within algorithm might not have a clear interpretation in a rolling-window environment when the planning horizon for the heuristics is smaller than the expected order cycle length. This is because in contrast to the static Wagner-Within algorithm the heuristics can not place orders which include the demand for too many periods ahead, while doing so would be optimal. The constrained Wagner-Within algorithm can provide a benchmark solution that does have meaning. In the constrained Wagner-Within algorithm, no order that fulfills demand for a number of periods longer than the planning horizon is allowed. This can be forced by modifying the earlier used expression for the Wagner-Within algorithm as follows:

$F(t) = \min\{Z_{kt} + F(k-1) | k = \Phi, ..., t\}$

where, given a rolling window length of n, $\Phi = 1$ if $t \leq n$ and $\Phi = t - n + 1$ otherwise. Obtaining the optimal schedule is done in the same way as with the static Wagner-Within algorithm.

	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100
Replenishments	180	0	275	0	0	150	0	250	0	250	0	0
Ending inventory	100	0	150	50	0	100	0	125	0	150	100	0
											Total costs:	$2375

Table 4: Results of applying the constrained Wagner-Within algorithm on the test case

2.1.3 Dynamic Wagner-Within

In a rolling-window environment, the Wagner-Within algorithm can also be used as a heuristic. This can be done by using only the information available for the length of the horizon, and applying the static Wagner-Within algorithm on this set of data.

2.2 Heuristics

We shall now explain and illustrate the working of various heuristics that will be used. The heuristics all work in a similar fashion. At a point t they consider the fixed costs K and the total holding costs H_t, which are defined as follows:

$$H_t = h \sum_{i=2}^{t} (i-1) d_i$$

The heuristics then apply their decision criterion based on these costs, and decide wether to start a new order in period 1 of size D_t, the accumulated demand over periods 1 to t:

$$D_t = \sum_{i=1}^{t} d_i$$

2.2.1 Part-Period Algorithm

The Part-Period Algorithm (PPA) is based on the earlier mentioned well-known property of the perfect balance between holding and ordering costs in the optimal solution of the EOQ. Although this property does not hold in general for the ELS problem, PPA tries to balance the holding costs and the ordering costs by placing an order whenever the holding costs are equal to or exceed the ordering costs. Thus, we set $Q_1 = D_t$ whenever

$$H_t \leq K < H_{t+1}$$

In our example, we see that

$$H_2 = \$100 < \$320$$

and that

$$H_3 = \$100 + 2 \times \$125 = \$350 > \$320$$

Therefore, the first order size will be D_2, covering the demand for the first two periods. Baker (1989) argues that modifying the stopping rule to place an order only when the holding costs exceed the ordering costs can alter the performance of this heuristic. However, the chances of exact equality are practically negligible in our data sets, so this shall not be considered.

	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100
Replenishments	180	0	275	0	0	150	0	250	0	250	0	0
Ending inventory	100	0	150	50	0	100	0	125	0	150	100	0
											Total costs:	\$2375

Table 5: Results of applying PPA on the test case

2.2.2 Least Total Cost

Another heuristic that tries to use the balance of the holding costs and the ordering costs in the EOQ to create a good solution for the ELS problem is Least Total Cost (LTC). It does this in a slightly more sophisticated way than PPA. Just like PPA, the heuristic searches for the first time that the holding costs exceed the ordering costs, and then determines the next order quantity as follows:

$$\begin{aligned} Q_1 &= D_t & \text{if } K - H_t < H_{t+1} - K \\ Q_1 &= D_{t+1} & \text{if } K - H_t \geq H_{t+1} - K \end{aligned}$$

That is, while PPA places an order as soon as the holding costs become equal to or larger than the fixed order costs, LTC also looks at the next periods' holding costs to get the holding costs and the order costs in the solution as close as possible. The difference that this alteration can make already becomes visible in the first order of our example. We have already seen that $H_2 = \$100$ and that $H_3 = \$350$. Where PPA would now set $Q_1 = D_2$, LTC finds that H_3 is closer to K than H_2 (or in terms of the notation above: $K - H_t \geq H_{t+1} - K$), and therefore chooses $Q_1 = D_3 = 305$.

	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100
Replenishments	305	0	0	300	0	0	0	350	0	0	150	0
Ending inventory	225	125	0	200	150	100	0	225	100	0	100	0
											Total costs:	$2505

Table 6: Results of applying LTC on the test case

2.2.3 Least Period Cost

Where PPA and LTC tried to exploit the EOQ's property of balanced holding and order costs in the optimal solution, Leat Period Cost (LPC) uses the property that "the total relevant costs per unit time for the duration of the replenishment quantity are minimized" (Silver et al., 1998). LPC is often referred to as the the Silver-Meal heuristic, for it were Silver and Meal (1973) who first proposed this algorithm. Since in our case the relevant costs are the ordering costs and the holding costs, the criterion becomes

$$\frac{K + H_t}{t}$$

and LPC places the first order up to the period where this first reaches a minimum, that is when

$$\frac{K + H_{t+1}}{t+1} > \frac{K + H_t}{t}$$

In our example, the ratio for period 1 becomes

$$\frac{K + H_1}{1} = \frac{\$320 + 0}{1} = \$320$$

The ratio for period 2 becomes

$$\frac{K + H_2}{2} = \frac{\$320 + \$100}{2} = \$210$$

Since this is lower than $320, we continue to calculate the ratio for period 3:

$$\frac{K + H_3}{3} = \frac{\$320 + \$350}{3} = \$223\frac{1}{3}$$

The ratio has a minimum at period 2, and therefore the heuristic chooses $Q_1 = D_2$. The results of the application on the test case can be found in Table 7. Note that this need not be the global minimum, but this might also be a local minimum. However, since in most real cases the chance of the ratio decreasing is small, we do not calculate the rest of the ratios after a local minimum has been found.

	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100
Replenishments	180	0	325	0	0	0	225	0	275	0	0	100
Ending inventory	100	0	200	100	50	0	125	0	150	50	0	0
											Total costs:	$2375

Table 7: Results of applying LPC on the test case

2.2.4 Least Unit Cost

Least Unit Cost (LUC) is a heuristic very similar to LPC, but instead of minimizing the costs per period, LUC uses minimization of the costs per unit as its goal. The criterion therefore becomes to set $Q_1 = D_t$ when

$$\frac{K + H_{t+1}}{D_{t+1}} > \frac{K + H_t}{D_t}$$

The outcome of using the LUC-algorithm on the test case can be found in Table 8.

	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100
Replenishments	305	0	0	200	0	0	225	0	225	0	150	0
Ending inventory	225	125	0	100	50	0	125	0	100	0	100	0
											Total costs:	$2425

Table 8: Results of applying LUC on the test case

2.2.5 H*

All the heuristics mentioned before were heuristics that have widely been used and discussed in textbooks and articles. The H* heuristic however, is a fairly new and unknown one. The H* heuristic was developed as a byproduct during research in the relation between the holding costs and the ordering costs in an optimal solution of the ELS model by Van den Heuvel and Wagelmans (2009). The reason that this heuristic is so distinguishable from most others, as mentioned earlier, is that the H* heuristic has a worst-case ratio of exactly 2. The H* heuristic tries to make the time between orders as long as possible, such that it is not possible to add an order which can result in a decrease in costs. In other words, the heuristic starts a new order in period t if there exists a $p \in \{1, ..., t-1\}$ so that

$$2K + h\sum_{i=2}^{p-1}(i-1)d_i + h\sum_{i=p+1}^{t}(p-1)d_i \leq K + h\sum_{i=2}^{t}(i-1)d_i$$

which can be simplified to

$$K \leq (p-1)h\sum_{i=p}^{t} d_i$$

where the right hand side can be interpreted as the holding cost savings by starting a new order in period p. So a new order should be set up if the savings in the holding costs exceed the increase in ordering costs for placing a second order in the interval. For example, when we take $t = 4$ en $p = 3$ in the test case, the savings in holding costs are $2 \times (\$100 + \$125)$, which exceeds the fixed order costs of \$320. Therefore, a new order is placed at the fourth period. From Table 9 it becomes clear that the H* heuristic indeed tends to create few orders.

	1	2	3	4	5	6	7	8	9	10	11	12
Demand	80	100	125	100	50	50	100	125	125	100	50	100
Replenishments	305	0	0	300	0	0	0	400	0	0	0	100
Ending inventory	225	125	0	200	150	100	0	275	150	50	0	0
										Total costs: \$2555		

Table 9: Results of applying H* on the test case

2.2.6 Modified heuristics

We can investigate our hypothesis by altering the heuristics in such a way that they obtain a worst-case ratio of two, and then compare the performance of these modified heuristics to the original ones. Van den Heuvel (2011) has found that any on-line heuristic that places their order at the same time or later than PPA and at the same time or earlier than H* has a worst-case ratio of exactly two. Thus we now have a way to alter any on-line heuristic so that it obtains a worst-case ratio of two. Let t denote the time at which the on-line heuristic would place the next order, and let t^{PPA} and t^{H^*} be the times at which PPA and H* would place the next order. Then t^*, the time at which the next order will be placed can be defined as follows.

$$t^* = \begin{cases} t^{PPA}, & \text{if } t < t^{PPA}. \\ t, & t^{PPA} \leq t \leq t^{H^*}. \\ t^{H^*}, & \text{if } t > t^{H*}. \end{cases} \tag{1}$$

Modifying the on-line heuristics in this way will result in heuristics with a worst-case ratio of exactly two. The performance of these new heuristics can then be compared with the original heuristics on the simulated data set of Simpson (2009). If our presumption is correct, the modified heuristics should perform better since they now have a worst-case ratio of two. We shall call these modified heuristics *LPC modified*, *LTC modified* and *LUC modified*.

3 Analysis

Before drawing any conclusions, it might be wise to verify that the heuristics have been implemented correctly. To assert the correctness of the implementation of the heuristics, we can run the heuristics on the test beds by Berry (1972) and Simpson (2009), and then compare our results with theirs. Our performance results can be found in Figure 1 and Table 10, where Y is the number of problems in which the procedure finds a suboptimal solution and Z is the average

	H*	PPA	LPC	LTC	LUC
Y	3	4	6	15	12
Z	0.664	1.339	0.943	10.274	9.303

Table 10: Performance of the heuristics on Berry's test bed

percent by which the costs corresponding to the heuristics' solution exceed the costs of the optimal solution. We see that our results do indeed match the results of the aforementioned articles. For the H* heuristic, the results match those as shown in Van den Heuvel and Wagelmans (2009).

3.1 Performance of the modified heuristics

The results of running the modified heuristics on the data set by Simpson (2009) can also be found in Figure 1. First of all, it is notable that LUC performs significantly better after the modification. The average cost gap between the optimal solution and LUC's solution was 13.1% at a horizon of 20 periods, while it is 6.3% when a worst-case ratio of 2 is forced upon it. LTC has improved as well, albeit a less impressive improvement than LUC has experienced; the average cost gap went down from 9.0% to 7.6%. LPC, already the best heuristic considered in this research, saw its performance deteriorate, although this was not a significant difference. The average cost gap of LPC went up from 4.1% to 4.4% at a horizon of 20 periods. Taken all together, it does seem to be the case that having a worst-case ratio of 2 has positive consequences for the performance of the heuristic.

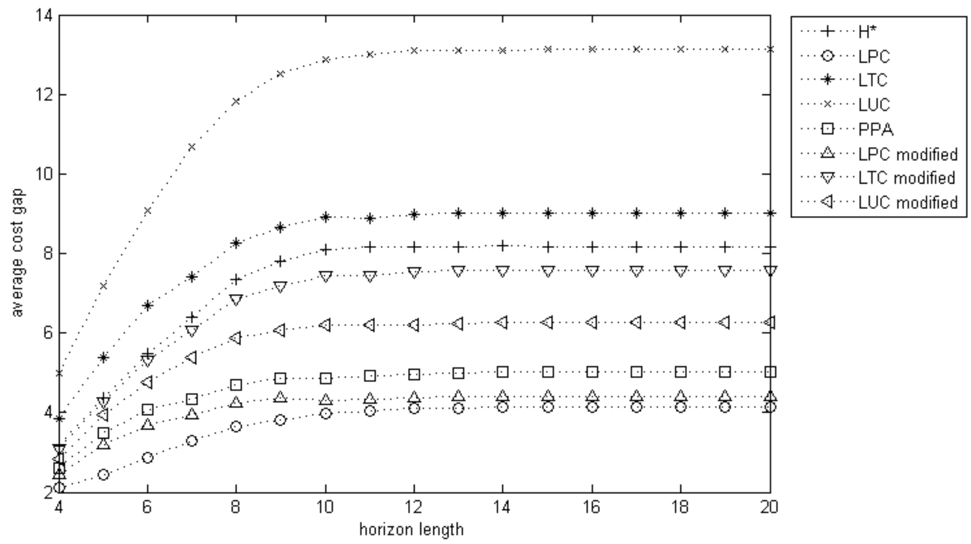

Figure 1: Performance of the heuristics on the test bed by Simpson (2009)

3.2 Influence of the test bed

The results above have been obtained by using the same test bed and the same type of analysis as Simpson (2009). It might however be the case that the results obtained above are very dependent on various aspects of this used test bed. In my opinion, there are a few shortcomings to this test bed, although it has to be said that constructing a representative test bed for the ELS problem is very difficult. In the following part, we shall look into the possible shortcomings of this test bed.

3.2.1 K/h-ratio

First of all, the expected cycle length in the used test bed ranges from 3 to 8, but the choice for this range is not elucidated and therefore the choice for this range seems a bit arbitrary. Also, all average cost gaps are aggregated over the different values for the expected cycle length. Of course, aggregating over some of the variables is necessary for the results to remain clear and interpretable, but sometimes important information is overlooked this way. For example, it might be the case that certain heuristics perform very well on test beds with low expected cycle lengths, and perform worse on test beds with higher values for the expected cycle length. It should be noted that using different expected cycle lengths corresponds to choosing different K/h-ratios, since the K/h-ratio in Simpsons test bed is:

$$K/h\text{-ratio} = \frac{\frac{T^2 D_i h}{2}}{h} = \frac{T^2 D_i}{2}$$

Where T represents the expected cycle length and D_i is the average demand over the entire planning horizon. It is known that the K/h-ratio can greatly influence the performance of heuristics. For example, Baker (1989) states that "the IOQ is a very effective procedure" but also that "the IOQ procedure is systematically flawed by a tendency to create suboptimal lot sizes when the ratio K/h is quite large."

We can test the influence of the K/h-ratio on the performance of the heuristics by aggregating over the rolling horizon length instead of aggregating over the expected cycle lengths in Simpsons test bed. The results of this can be found in Figure 2. Since the behavior for expected cycle lengths longer than 8 might also be interesting, we vary the expected cycle length from 3 to 12. It becomes immediately clear that the K/h-ratio used has consequences for the performance of the heuristics. When the K/h-ratio is low, the heuristics average cost gap is scattered between 1% and 13%. However, when the K/h-ratio is relatively high, the average cost gap of all heuristics seems to focus at a very small interval. Perhaps the most interesting fact is that the on-line heuristic with the best performance for small K/h-ratios, LPC, has by far the worst performance when that ratio is high, so *LPC modified* performs better for high values of the expected cycle length. Furthermore, *LTC modified* performs better than LTC up until an expected cycle length of 5, and has equal performance from there onwards. Our data show that when the expected cycle length increases, *LTC modified* reschedules less orders (i.e. plan an order earlier or later than LTC would plan it because of our modification rule); about 13.3% for a cycle length of 3 against 0.4% for cycle lengths of 8.

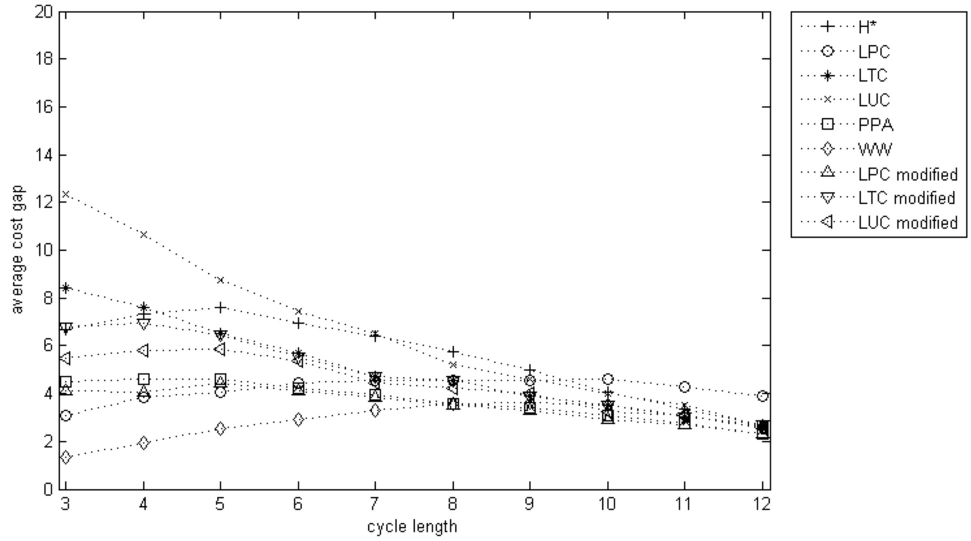

Figure 2: Performance of the heuristics over different cycle lengths on the test bed by Simpson (2009)

3.2.2 Coefficient of variation

But the results for this test bed were not only aggregated over the expected cycle length, they were also aggregated over the three different parts of the test bed. The three different parts all had different coefficients of variation, and therefore different amounts of periods with zero demand. Again, important information may be overlooked by doing so. We can test the influence of the variability of the demand in a similar fashion as we did for investigating the influence of the K/h-ratio. Since Simpsons test bed only contains three different values for the coefficient of variation however, we shall create a new test bed. The new test bed consists of ten parts and is constructed in the same way as Simpsons original test bed, with a standard deviation of 1,000 and means varying from 1,000 to 10,000, so the squared coefficient takes values from 0.1 to 1.0. We can then apply the heuristics to this test bed, aggregating over the horizon lengths and expected cycle lengths as before.

The performance of the heuristics on test beds with different coefficients of variation can be found in Figure 3. Although it becomes clear that the H* heuristic is the most robust against different coefficients of variation in the data, its performance is very bad on test beds with low amounts of variation in comparison with the other heuristics. If we ignore the H* heuristic, we see a similar pattern as with the expected cycle lengths before; for low values of variation, the heuristics all perform similarly well, but the performance becomes more diverse for high coefficients of variation. As before, *LUC modified* performs better than LUC, but *LTC modified* only performs significantly better than LTC for high coefficients of variation. *LPC modified* performs worse than LPC for all coefficients of variation.

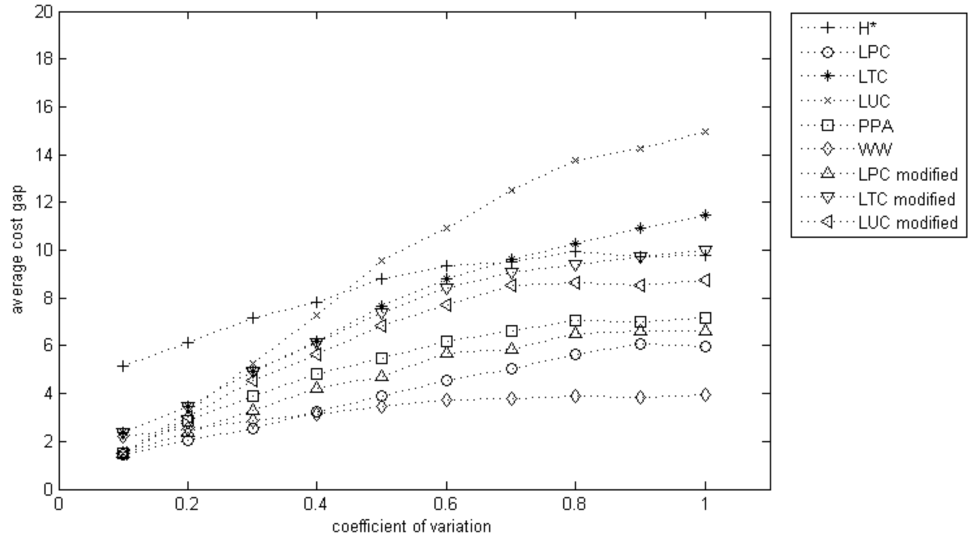

Figure 3: Performance of the heuristics on test beds with different coefficients of variation

3.2.3 Skewness

The previous investigation into the shortcomings of the used test bed should be sufficient to give a conclusion regarding our hypothesis and to evaluate the reliability of this conclusion. However, there are more aspects to the test bed which can affect the performance of heuristics and which are therefore interesting to look into. One of those are properties of the used distribution, in this case the normal distribution. While the normal distribution is symmetric, it is also interesting to examine the behavior of the heuristics under demand patterns which come from a skewed distribution. To investigate this, we shall create a new test bed with demand patterns sampled from the Gamma distribution with a standard deviation of 1,000 and with different values for the skewness. However, when the skewness of a distribution changes, so do the variance and the mean, which influences the coefficient of variation which then influences the performance of the heuristics, as seen before. It is therefore important that we can link the skewness of the distribution to the coefficient of variation, so we can filter out the effects of the changed coefficient of variation on the performance on the heuristics. For the Gamma distribution, we know the following; when $X \sim GAM(\theta, k)$

- $E(X) = k\theta$
- $Var(X) = k\theta^2$
- $Skewness = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{2}{\sqrt{k}}$

For a given value of skewness, γ and a given mean μ, we can then express the distribution parameters k and θ;

$$k = \left(\frac{2}{\gamma}\right)^2$$

$$\theta = \frac{\mu}{k} = \frac{\mu}{\left(\frac{2}{\gamma}\right)^2}$$

Which we can then use this to express the coefficient of variation in terms of the skewness as follows;

$$c_v = \frac{\sigma}{\mu} = \frac{\sqrt{k\theta^2}}{k\theta} = \frac{\sqrt{k}\theta}{k\theta} = \frac{1}{\sqrt{k}} = \frac{\gamma}{2}$$

Thus, theoretically the skewness is precisely twice the coefficient of variation, which is a very nice property for our research. If we take values of 0.2 to 2.0 with steps of 0.2 for the value of the skewness, the coefficient of variation automatically takes values from 0.1 to 1.0 with steps of 0.1. The effect of this factor on the performance of the heuristics was already found in the previous section, and thus we can filter out the effects of the varying coefficient of variation on performance when we want to find the effect of the varying skewness.

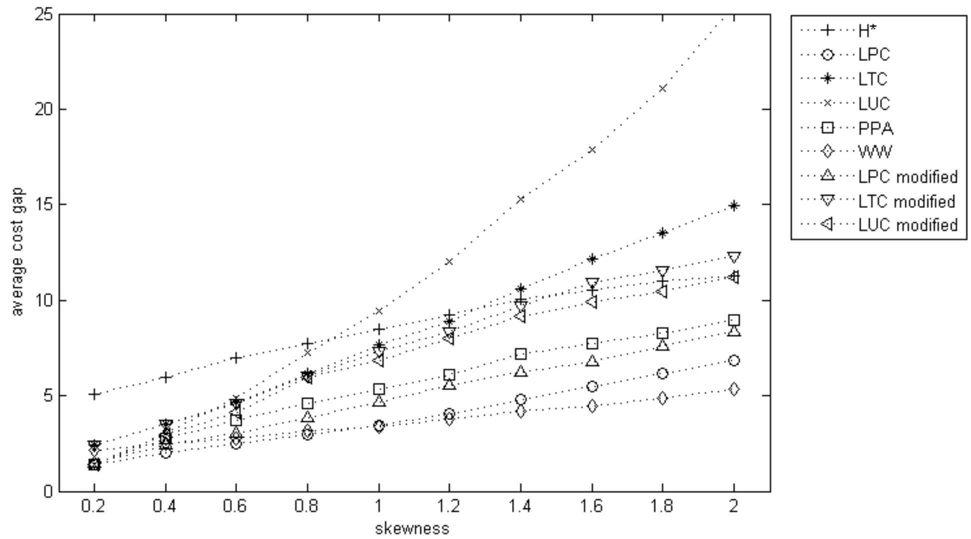

Figure 4: Performance of the heuristics on test beds with different skewness values

The results of applying the heuristics on the new test bed are displayed in Figure 4. Note that one could also choose to place the coefficient of variation on the horizontal axis with values from 0.1 to 1. It seems that varying the skewness has a very similar effect as just varying the coefficient of variation, which is no surprise as increasing the skewness corresponds with increasing the coefficient of variation, as mentioned before. However, the performance of the heuristics seems to deteriorate a bit more when one skews the distribution instead of just increasing the coefficient of variation of the distribution. This becomes more clear when we take the difference between the performance of the heuristics in Figure 4 and Figure 3, which is shown in Figure 5. Since all factors between Figure 4 and Figure 3 are equal, except for the way in which the coefficient of variation is increased, Figure 5 represents the increase in the average cost gap when the distribution is skewed. It is clear that the skewed distribution has a negative effect on the performance of the heuristics, although not every heuristics' performance deteriorates equally fast with an increase of the skewness. However, *LUC modified* and *LTC modified* perform better than LUC and LTC when the distribution is skewed.

However, we used Gamma distributions for simulating the skewed test bed, while we used normal distributions for the original test beds. While the mean and variance were kept equal for every part between the original and the skewed test beds, it might be the case that the change in performance was caused by some other property of the distributions that has changed besides the skewness. One other attribute that always differs between the normal and the Gamma distribution is the excess kurtosis, which is 0 for the normal distribution and $6/k$ for the Gamma distribution. However, this increase of kurtosis is caused by the skewness of the distribution, so any influence this has on the change in performance can also be allocated to the change in skewness. To my knowledge, there is no other important factor left that can have caused the deterioration of the performance of the heuristics. More research into this might be useful to strengthen the conclusion.

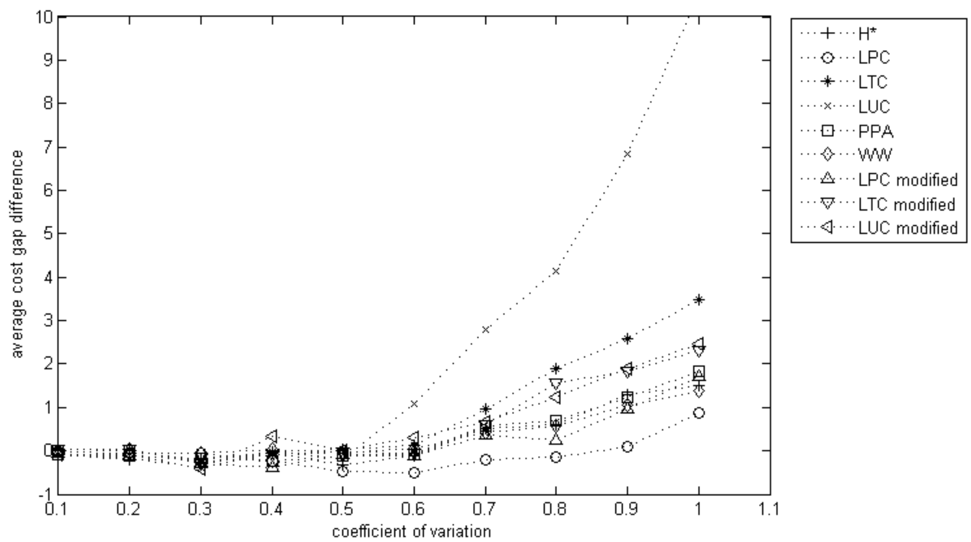

Figure 5: The difference between skewed and symmetric distributions on performance

4 Conclusion

In this thesis, we have tried to determine whether the property of having a worst-case ratio of 2 is a good property for an on-line heuristic to have. We have done so by using PPA and H* to modify LPC, LTC and LUC in a way that forces a worst-case ratio of 2 upon them, and then comparing the performance of the modified heuristics with the performance of the original ones. It is very difficult to provide a straightforward and unambiguous conclusion. From the first analysis, which was done similarly to Simpson (2009), one would say that having a worst-case ratio of 2 is indeed a good property for an on-line heuristic to have, since *LUC modified* and *LTC modified* both performed significantly better than their unmodified counterparts.

However, the test bed used seems to be of influence on the results. For low expected cycle lengths, *LTC improved* performs better than LTC, and *LPC modified* performs worse than LPC, while for high expected cycle lengths all heuristics but LPC perform equally well, and LPC performs worse. The coefficient of variation also plays a role in drawing our conclusions. For low coefficients of variation, all heuristics perform very well, with the exception of H*, which performs very badly. For higher coefficients of variation, the performance of the heuristics diverges, but again *LPC modified* always performs worse than LPC.

In general, LUC and LTC seem to benefit from the worst-case ratio of 2, while this property seems to impair the performance of LPC. Therefore, one cannot state that the possession of a worst-case ratio of 2 is a good property for an on-line heuristic to have in general.

While we were primarily interested in the performance of heuristics with worst-case ratios of 2, we also found a nice result regarding the performance of the original heuristics. When one is concerned with solving dynamic economic lot-sizing problems with the use of the heuristics described in this paper, the best results are almost always obtained by using LPC. LPC's performance is more robust against changes in the K/h-ratio, in the coefficient of variation and in the skewness than the other heuristics. The only situation in which one should consider the use of another heuristic is when the expected cycle length takes on high values.

References

S. Axsäter. Worst case performance for lot sizing heuristics. *European Journal of Operations Research*, 9:339–343, 1982.

K.R. Baker. Lot sizing procedures and a standard data set: A reconciliation of the literature. *Journal of Manufacturing and Operations Management 2*, pages 199–221, 1989.

W.L. Berry. Lot sizing procedures for requirements planning systems: A framework for analysis. *Production & Inventory Management*, 2nd quarter:19–34, 1972.

G.R. Bitran, T.L. Magnanti, and Yanasse H.H. Approximation methods for the uncapacitated lot size problem. *Management Science*, 30:1121–1140, 1984.

E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling*. Wiley, third edition, 1998.

E.C. Silver and H.C. Meal. A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, 2nd quarter:64–74, 1973.

N.C. Simpson. Questioning the relative virtues of dynamic lot sizing rules. *Computers & Operations Research*, 28:899–914, 2009.

W. van den Heuvel. A holding cost bound and a new heuristic for the economic lot-sizing problem. In *International Workshop on Lot Sizing*, 2011.

W. van den Heuvel and A.P.M. Wagelmans. A holding cost bound for the economic lot-sizing problem with time invariant cost parameters. *Operations Research Letters*, 37:102–106, 2009.

W. van den Heuvel and A.P.M. Wagelmans. Worst-case analysis for a general class of online lot-sizing heuristics. *Operations Research*, 58:59–67, 2010.

H.M. Wagner and T.M. Within. Dynamic version of the economic lot size model. *Management Science*, 5(No. 1):89–96, 1958.