Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis International Bachelor of Economics and Operations Research

Enhancement of the static-dynamic uncertainty strategy in lot-sizing via semi-static-dynamic heuristic

Zonne Heijnsbroek (523835)

Supervisor:	Heuvel, W van den
Second assessor:	Wagelmans, APM
Date final version:	21st July 2023

Abstract

The lot-sizing problem is a well-known optimization problem which aims to determine production and inventory decisions. This thesis addresses the stochastic lot-sizing problem under static-dynamic uncertainty and introduces a novel approach called the semi-static-dynamic uncertainty strategy. We begin by replicating the results of a mixed-integer programming (MIP) model proposed by Tunc, Kilic, Tarim and Rossi (2018) for the static-dynamic uncertainty strategy with α service level constraints. Next, we present the semi-static-dynamic uncertainty strategy, which combines elements of both the static-dynamic and fully dynamic uncertainty strategies. This innovative heuristic algorithm allows for adjustments to the initial replenishment schedule based on real-time demand information, improving responsiveness to demand fluctuations and generating cost savings. The semi-static-dynamic uncertainty strategy offers potential for enhancing the lot-sizing problem in real-world applications.

1 Introduction

The lot-sizing problem is a classic optimization problem that has been widely studied and applied in various industries, especially in manufacturing and supply chain management. The core of the lot-sizing problem involves determining the optimal production and inventory decisions to minimize the total cost over a discrete planning horizon, considering factors such as production, holding, setup, and back-ordering costs. The stochastic lot-sizing problem has attracted significant attention from both researchers and companies.

Previous studies have proposed several solution approaches for the stochastic lot-sizing problem, including stochastic programming, dynamic programming, and heuristic methods. However, many of these are hard to implement for large-scale, real-world problems. To tackle this issue, mixed-integer programming (MIP) can be used as a powerful and flexible tool for modeling and solving the stochastic lot-sizing problem. Nevertheless, the complexity of the problem and the large number of decision variables and constraints in the MIP formulations often lead to computational challenges.

In this thesis, we extend the paper of Tunc et al. (2018) by modelling the static-dynamic uncertainty lot-sizing problem with an extended mixed-integer programming formulation. We will replicate the results of the model with a constraint that ensures the service level. The static-dynamic uncertainty strategy determines the replenishment schedule at the beginning of the time horizon, while the order quantity is determined dynamically based on the current level of inventory. Furthermore, we present an innovative algorithm for the semi-static-dynamic uncertainty problem. This is a heuristic which blends the static-dynamic and dynamic uncertainty strategies. Benefits include superior adaption to demand fluctuations and cost savings. The main contribution of this thesis lies in the verification of previous results and the exploration and implementation of the semi-static-dynamic uncertainty strategy.

The remainder of this thesis is organized as follows: Section 2 provides a Literature Review of the stochastic lot-sizing problem and its existing solution methods. Section 3 provides the Problem Definition, whereupon Section 4 outlines the Methodology. The Results are presented in Section 5, after which Section 6 presents the Conclusion.

2 Literature review

In this Section we will present a literary review about the stochastic lot-sizing problem, policy's and solution methods.

2.1 Stochastic lot-sizing problem

In the stochastic lot-sizing problem, the aim is to determine the optimal production and inventory decisions such that the total expected costs are minimized. We will consider ordering costs, holding costs, penalties for back-orders, and the required service level. Further explanation of the problem can be found in Section 3.

Bookbinder and Tan (1988) discuss three policies on lot-sizing policy. Firstly, they discuss the **static uncertainty** model, where all decisions are made at the beginning of the time horizon. This approach is the most conservative but noticeably straightforward. Secondly, they discuss the **dynamic uncertainty** model where the decision whether to order (and how much) is decided at every point in time. Even though this approach is cost optimal, Bookbinder and Tan (1988) do not recommend this model in practice, since it is complex to solve and requires great flexibility in real life application. On the intersection of these strategies lies the **static-dynamic uncertainty** strategy, where a replenishment schedule is created at the beginning of the time horizon. The order quantity will be determined dynamically based on the current level of inventory. The static determination of the replenishment schedule ensures convenient scheduling of resources such as transport and human resources. The dynamic determination of the order quantity accounts for variable demand, since the order quantity is determined based on the current stock level instead of the expected stock level.

2.2 Order up-to policy

With a given replenishment schedule, we must set a policy to determine the amount that is ordered. Özen, Doğru and Tarim (2012) have reviewed the order policy for the static-dynamic uncertainty strategy problem. They found that in a problem with holding costs and penalty costs for service level, a base-stock policy is optimal. First, the base-stock level is determined by balancing the holding costs and penalty costs while maintaining a certain service level. At the scheduled order time, the difference between the base-stock level and the current stock will determine the amount ordered. The base-stock level can thus be considered an order-up-to level. By maintaining an optimal base-stock level, the policy balances the trade-offs between holding costs and penalty costs, while meeting the desired service level targets. The joint search for the optimal ordering schedule and order-up to level is computationally exhaustive and only applicable for small planning horizons, therefore it is advisable to use heuristics. (Özen et al., 2012)

2.3 Solution methods

Especially for larger planning horizons, computational complexity is a limiting factor in determining the optimal ordering schedule and order-up-to level. The heuristics used for this problem often fall into two categories: tailor-made algorithms and mathematical programming solvers. Tailor-made algorithms are designed specifically for a particular problem. They use problem-specific information to decrease the domains of decision variables. There is an abundance of literature on this topic, examples include state space augmentation (Rossi, Tarim, Hnich & Prestwich, 2011), branch-and-bound procedures (Tarim, Dogru, Ozen & Rossi, 2011), and dynamic programming (Özen et al., 2012).

Although tailor-made algorithms can provide a quick solution to a specific problem, they are not easily generalized and the solutions may be sub-optimal. For that reason, we prefer to use a mathematical programming model. A mathematical program is a more general tool to solve an optimization problem, formatted as a mixed-integer programming (MIP) model. They are applicable to a wider range of problems due to the adaptability of the model. Additionally, the solution quality is easily satisfied by being able to provide solutions within a specific optimality gap. Considering that MIP models are primarily designed to handle linear relationships, challenges emerge when encountering a non linear objective. In the lot-sizing problem the loss function is considered to be non-linear due to the combination of fixed and variable costs. Tarim and Kingsman (2004) have solved this problem by simply approximating the cost function as a linear function. However, this is only useful for high values of the required service level (denoted by α), since the model relies on omitting the back-orders and shortage costs. Another approach is using a piecewise linear relaxation of the cost function as suggested by Tarim and Kingsman (2006). By using four breakpoints, they were able to approximate the cost function with a worst-case error of 3.92%. This result can be further improved by increasing the number of breakpoints. The model of Tarim and Kingsman (2006) also incorporates the shortage costs, which were not considered in Tarim and Kingsman (2004) and Bookbinder and Tan (1988). This ensures the model will be valid for all required service levels. The model of Tarim and Kingsman (2006) is well-acknowledged as a benchmark model for this problem. Continuing on the piecewise linear approximation of the costs function, Rossi, Kilic and Tarim (2015) provide an extension where unmet demand is back-ordered or lost. They establish a framework that can solve different variants of the lot-sizing problem.

Tunc, Kilic, Tarim and Eksioglu (2014) present a MIP formulation with a network flow structure. They have accomplished to obtain a stronger linear relaxation compared to the benchmark model while also improving the computational performance. However, they have only applied this formulation to the α service level variant and do not explore other variants of the problem in this paper. Fortunately, an extended formulation has been developed that combines the research Tunc et al. (2014) and Rossi et al. (2015). The extended formulation of Tunc et al. (2018) inherits the flexibility of the Rossi et al. (2015) formulation, while also preserving the computational efficiency of the Tunc et al. (2014) flow formulation.

Besides presenting an extended formulation, Tunc et al. (2018) also presents a dynamic cut procedure to bypass the need for a pre-defined linear approximation of the cost function. With the dynamic cut procedure, we use a relaxed version of the problem to which we add cuts until the approximated loss function becomes sufficiently close to the actual loss function.

3 Problem definition

The stochastic lot-sizing problem for the static-dynamic uncertainty strategy aims to determine the optimal replenishment schedule and order quantities under uncertain demand. Demand is assumed to be variable over time, while other problem parameters such as costs and lead times are assumed to be constant. The objective of the problem is to minimize the total expected costs.

There are many variants of this problem with varying complexity. Some of these variants include *penalty costs*, which are incurred when customer demand is not met. The unmet demand can be either *back-ordered*, when customers will wait until their order is fulfilled, or *lost*, when customers will not wait for their order. An additional way to deal with customer satisfaction is to include service level constraints. The service level ensures that a certain percentage of customers is served without running out of stock.

The costs in this problem consist of fixed costs and variable costs. The ordering costs are considered to be fixed and are incurred every time an order is placed. The holding and penalty costs are considered variable and depend on the amount of inventory and the holding period of that inventory. As discussed in Section 2, the non-linearity of the costs function complicates the optimisation of the lot-sizing problem.

In this thesis, we will replicate the results of the extended model of Tunc et al. (2018), which entails a back-ordering inventory policy with penalty cost. We will include the α service level constraint to limit the occurrence of stock-outs. We will compare the performance statistics of our replication to the results of Tunc et al. (2018). Further explanation of the methods to be used can be found in Subsection 4.1.

Next to replicating and verifying the results of Tunc et al. (2018), we will introduce the semi-static-dynamic uncertainty strategy. The semi-static dynamic uncertainty strategy offers a novel approach to address the challenges of the stochastic lot-sizing problem within the context of static-dynamic uncertainty. Traditionally, the static-dynamic uncertainty strategy fixes the replenishment cycles at the beginning of the planning horizon, without considering the fluctuations in demand. However, this approach can lead to sub-optimal decisions when the demand significantly differs from the expected values. By re-evaluating the initial solution and making necessary adjustments, the semi-static-dynamic uncertainty strategy offers increased responsiveness to changing demand patterns while minimizing the computational complexity associated with a fully dynamic approach. With the heuristic presented in Subsection 4.2, we aim to further enhance the adaptability and efficiency of the lot-sizing problem by incorporating real-time demand information and optimizing the replenishment cycles iteratively.

4 Methodology

This Section explains the research methodology, which consists of two parts. First, we introduce the extended Mixed-Integer programming formulation of Tunc et al. (2018) with the α service level constraint in Subsection 4.1. We refer to this model as the PM model. Secondly, we will discuss the semi-static-dynamic uncertainty strategy in Subsection 4.2.

4.1 PM model

In the PM model, we desire a solution for the lot-sizing problem with static-dynamic uncertainty strategy. This consists of two components: the replenishment schedule and a base-stock level. The objective is to minimize the expected total costs over the entire planning horizon of N periods. In addition to the basic PM model of Tunc et al. (2018), we also include the α service level constraint. We assume unmet demand is back-ordered.

Let D_{ij} denote the random demand over a given time interval $[i, j)$ with cumulative distribution function $F_{ij}(\cdot)$, and first-order loss function $L_{ij}(\cdot)$. We consider three types of costs: holding costs h (costs of each unit of inventory that is carried from a period to the next), back-order costs p (back-order costs per unit of back-ordered demand per period), and a fixed setup cost K (incurred for each order).

The periods $T_1, ..., T_m$ represent the replenishment periods over the planning horizon. T_n refers to the period in which the nth order is scheduled. The number of replenishment periods is thus given by m. A replenishment cycle is defined as any interval between successive replenishment periods, $[T_n, T_{n+1})$. For notational brevity, we assume $T_1 = 1$ and $T_{n+1} = N + 1$. We can now view the planning horizon as the set of m planning periods. The postreplenishment inventory level is denoted by y, which gives the inventory level at the beginning of a cycle after the order is received.

Similar to Tunc et al. (2018), we approximate the loss function $L_{ij}(\cdot)$ with a piecewise linear approximation denoted by $L(y)$. The approximation of the piecewise linear approximation is constructed according to the paper of Rossi, Tarim, Prestwich and Hnich (2014). The approximation is given by $L(y) \approx \max_{(a,b)\in W_{it}}\{a + by\}$ for a finite set of intercept and slope pairs $W = \{(a_1, b_1), (a_2, b_2), ...\}$. The lower bound for the piecewise linear function with 11 segments is given by $\hat{L}_{lb}(x; \omega) = \sum_{i=1}^{N} p_i \max(x - \mathbb{E}[x|\Omega_i], 0)$, where the parameter values of p_i and $\mathbb{E}[x|\Omega_i]$ can be found in the paper of Rossi et al. (2014). p_i refers to the weight associated with each segment of the piecewise linear approximation, reflecting the probability of the inventory level falling within a certain range. $\mathbb{E}[x|\Omega_i]$ represents the expected value of x given the state Ω_i, i.e., the expected inventory level within a specific range. The following function is equivalent to the lower bound of the first order loss function and is used to compute the values of $W = \{a, b\}$:

$$\hat{L}_{lb}(x;\omega) = \begin{cases} 0, \text{ if } 1 \leq x \leq \mathbb{E}[\omega|\Omega_1] \\ p_1 x - p_1\mathbb{E}[x|\Omega_1], \text{ if } \mathbb{E}[\omega|\Omega_1] \leq x \leq \mathbb{E}[\omega|\Omega_2] \\ (p_1 + p_2)x - (p_1\mathbb{E}[\omega|\Omega_1] + p_2\mathbb{E}[\omega|\Omega_2]), \text{ if } \mathbb{E}[\omega|\Omega_2] \leq x \leq \mathbb{E}[\omega|\Omega_3] \\ \vdots \\ (p_1 + p_2 + \ldots + p_N)x - (p_1\mathbb{E}[\omega|\Omega_1] + p_2\mathbb{E}[\omega|\Omega_2] + \ldots + p_N\mathbb{E}[\omega|\Omega_N]), \text{ if } \mathbb{E}[\omega|\Omega_{N-1}] \leq x \leq \mathbb{E}[\omega|\Omega_N] \end{cases}$$

This approximation is based on a normal distribution. The standardization of the non-standard loss function is performed according to Lemma 7 of Rossi et al. (2014). The first order loss function of the normally distributed variable ξ can be expressed in terms of the standard normal function as: $\hat{L}(x;\xi) = \sigma\hat{L}\left(\frac{x-\mu}{\sigma};Z\right)$, where μ and σ are the mean and standard deviation of ξ, and Z is a standard normal random variable. Using this property, the standardized values for $W = \{a, b\}$ are found.

We continue with the PM model and introduce the following decision variables:

- x_{ij} is 1 if $[i, j)$ is a replenishment cycle, and 0 otherwise;
- q_{ij} is the expected cumulative order quantity up-to and including period i if $[i, j)$ is a replenishment cycle, and 0 otherwise;
- H_{ijt} is the approximate loss function value at period t of replenishment cycle $[i, j)$.

The replenishment schedule can be derived from the values of x_{ij}. The base-stock level of the first period of replenishment cycle $[i, j)$ is $q_{ij} - \mathbb{E}D_{1i-1}$. We also introduce constraints that ensure that if $x_{ij} = 0$, both q_{ij} and H_{ijt} will be zero as well.

The objective function to minimize costs is formulated as follows:

$$\min \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} (Kx_{ij} + \sum_{t=i}^{j-1} (h(q_{ij} - \mathbb{E}D_{1t}x_{ij}) + (h+p)H_{ijt})) \tag{1}$$

The expression inside the outer summations gives the expected costs over the interval $[i, j)$, which is composed of the setup costs K, holding costs h, and back-order cost p. If the interval $[i, j)$ is not a replenishment cycle, this expression will be zero. The total costs are determined by the summation over all disjoint replenishment cycles.

$$\text{s.t} \quad \sum_{i=1}^{t-1} x_{it} = \sum_{j=t+1}^{N+1} x_{tj}, \quad t \in [2, N] \tag{2}$$

$$\sum_{j=2}^{N+1} x_{1j} = 1 \tag{3}$$

$$\sum_{i=1}^{N} x_{i(N+1)} = 1 \tag{4}$$

The establishment of the planning horizon as a disjoint union of replenishment cycles is secured by constraints (2), (3), and (4). The constraints are formulated as common flow equations, where the periods represent nodes and the replenishment cycles represent arcs. Constraint (2) ensures that if a replenishment cycle starts in period t, another replenishment cycle must end at period t as well, for all $t \in [2, N]$. Constraint (3) ensures that a replenishment cycle starting at the first period will end in some other period. Constraint (4) ensures the last replenishment cycle ends at the last period.

$$q_{ij} \leq Mx_{ij}, \quad i \in [1, N],\ j \in [i+1, N+1] \tag{5}$$

To manage the relationship between $x_{ij} = 0$ and q_{ij}, constraint (5) is employed. It ensures that if $x_{ij} = 0$, q_{ij} is zero as well. The value of M is chosen to be sufficiently large. We use the bound of the inverse distribution function at the end of the planning horizon, evaluated at the critical percentile $p/(h+p)$

$$\sum_{i=1}^{t-1} q_{it} \leq \sum_{j=t+1}^{N+1} q_{tj}, \quad t \in [2, N] \tag{6}$$

Next, constraint (6) ensures that the cumulative order quantities are non-decreasing.

$$H_{ijt} \geq a x_{ij} + b(q_{ij} - \mathbb{E}D_{1(i-1)} x_{ij}), \quad \begin{array}{l} i \in [1, N], \quad j \in [i+1, N+1], \\ t \in [i, j-1], \quad (a, b) \in W_{it}. \end{array} \tag{7}$$

Constraint (7) ensure that for every replenishment cycle $[i, j)$, H_{ijt} is larger then the value of the loss function evaluated at the base-stock level, given by $L_{it}(q_{ij} - \mathbb{E}D_{1(i-1)} x_{ij})$. Since we approximate $L(\cdot)$ via piecewise linear approximation, $L(y) \approx \max_{(a,b) \in W_{it}} \{a + by\}$ for a given finite set of intercept and slope pairs $W = \{(a_1, b_1), (a_2, b_2), ...\}$. We then obtain $L_{it}(q_{ij} - \mathbb{E}D_{1(i-1)}) \approx \max_{(a,b) \in W_{it}} \{a + b(q_{ij} - \mathbb{E}D_{1(i-1)}\}$.

In addition to the preceding constraints, we incorporate a specific measure of service level into the formulation. This service level is of importance when not only the cost efficiency, but also the customer satisfaction is considered. Specifically, we impose an α service level constraint, setting a lower bound α on the non-stockout probability for every period across the planning horizon. The α service level constraint is formulated in constraint (8),

$$q_{ij} \geq (\mathbb{E}D_{1(i-1)} + F^{-1}_{i(j-1)}(\alpha)) x_{ij}, \quad i \in [1, N], \quad j \in [i+1, N+1] \tag{8}$$

This constraint should hold in every period on the planning horizon, but since in every replenishment cycle $[i, j)$ the non-stockout probability increases, it is sufficient to apply this constraint to the last period of every replenishment cycle (period $j-1$ of cycle $[i, j)$). The inventory level at period $j-1$ is given by $(q_{ij} - \mathbb{E}D_{1(i-1)}) - D_{i(j-1)}$. The non-stockout probability is then $\Pr\big((q_{ij} - \mathbb{E}D_{1(i-1)}) - D_{i(j-1)} \geq 0\big) \geq \alpha$, equivalently given by $F_{i(j-1)}(q_{ij} - \mathbb{E}D_{1(i-1)}) \geq \alpha$. Rewriting gives us the equation $q_{ij} \geq (\mathbb{E}D_{1(i-1)} + F^{-1}_{i(j-1)}(\alpha)) x_{ij}$. The constraint is only enforced when $[i, j)$ is a replenishment cycle by the term x_{ij} in constraint (8).

$$H_{ijt} \geq 0 \quad i \in [1, N], \ j \in [i+1, N+1], \ t \in [i, j-1], \tag{9}$$

$$q_{ij} \geq 0, \quad x_{ij} \in \{0, 1\} \quad \forall i \in [1, N], \ j \in [i+1, N+1], \tag{10}$$

Lastly, the bounds of the decision variables are given in constraints (9) - (10). The PM model with service level constraint is then given by objective (1) subject to (2) - (10).

4.2 Semi-static-dynamic uncertainty strategy

In this subsection, we delve into the semi-static-dynamic uncertainty strategy. First, we introduce the concept and operational structure of this strategy. Secondly, we elaborate on different adjustment conditions, after which a small-scale numerical example shows a practical demonstration of the potential benefits of this strategy.

4.2.1 Introduction semi-static-dynamic algorithm

In the static-dynamic uncertainty strategy, scheduled order times are fixed and do not respond to demand fluctuation. Consider a scenario where an order is scheduled but the demand has dropped significantly. It could be inefficient to place an order, since the order quantity is small and not worth the fixed order costs. Also, an environmental argument can be made, since deliveries made by near-empty vehicles lead to inefficient fuel usage and unnecessary carbon emissions. To address this inefficiency, we present an algorithm that proposes a cross-over between the static-dynamic uncertainty strategy and dynamic uncertainty strategy. It aims to better adapt to real world demand fluctuations, while not being as computationally exhaustive as the full dynamic strategy.

In the static dynamic uncertainty strategy, the replenishment schedule is determined at the beginning of the time horizon. The amount that is ordered is decided at start of a replenishment cycle. There is thus one decision moment that fixes the replenishment cycles. In the fully dynamic uncertainty strategy, the decision whether to order and the amount that is ordered, is made in every time period. We propose a heuristic which re-evaluates the solution found at the beginning of the planning horizon. The algorithm solves the static-dynamic problem initially and allows for adjustments during the order periods, transforming the problem from static-dynamic to **semi**-static-dynamic.

An essential feature of this algorithm is its inherent flexibility both in terms of adjustment conditions and the adjustment process itself. The *adjustment condition* triggers a re-evaluation and possible modification of the initial solution under certain circumstances. While our paper focuses on the adjustment condition of lower-than-expected demand, the algorithm can accommodate a variety of adjustment conditions, tailored to the specific needs or uncertainties of any given scenario. Furthermore, the *adjustment* can be interchanged and modified according to different operational needs. The structure of the algorithm allows for the implementation of various adjustment methodologies, thus making it versatile and adaptable to a broad range of situations.

The algorithm works as follows:

1. **Initialization**: Solve the PM model to obtain the original replenishment cycles.

2. **Adjustment evaluation**: For each order period after the first, evaluate whether an adjustment might be beneficial according to the *adjustment condition*.

3. **Adjustment**: When an order period fulfills the *adjustment condition*, exclude this period from being an order period. Solve the PM model again from the next period onward. Start the adjustment evaluation from the next order period.

4. **Evaluation**: Once all order periods have been evaluated and the possible adjustments have been performed, the realized objective value is computed using the realized demands.

The realized objective values is compared for the original replenishment cycles and the adjusted replenishment cycles to find the effect of adjustment.

4.2.2 Adjustment condition

In order to determine when a re-evaluation of the replenishment schedule should be made, an adjustment condition is imposed based on the realized demand, denoted by $\hat{D}_{ij}$. The focus in this thesis is on cases where the realized demand is lower than expected. The goal is to establish a criterion that evaluates the adequacy of the order schedule that was determined using information available at the beginning of the planning horizon.

The parameter γ determines the sensitivity of the adjustment conditions. A larger γ factor will result in more frequent adjustments, potentially leading to improved responsiveness to demand fluctuations. However, it is important to strike a balance, as too many adjustments may not be practical in real-life situations due to associated costs or operational constraints.

One straightforward adjustment condition is to measure the difference between the realized and expected demand in the time period since the last order period. We denote the current period, in which the adjustment condition is evaluated, by v and the previous order period by u. Comparing the values of the expected demand in the last order period $\mathbb{E}D_{uv-1}$ with the realized demand $\hat{D}_{uv-1}$ gives us the following condition:

$$\text{If } (\hat{D}_{uv-1} \leq \gamma \cdot \mathbb{E}D_{uv-1}\ ), \text{ perform adjustment.} \tag{11}$$

This condition, expressed in Equation (11), compares the expected demand in the last order period $\mathbb{E}D_{uv-1}$ with the realized demand $\hat{D}_{uv-1}$. If the realized demand falls bellow the threshold set by γ, an adjustment is made. This condition evaluates the adequacy of the order schedule based on the demand realizations within the previous replenishment cycle.

$$\text{If } (\hat{D}_{1v-1} \leq \gamma \cdot \mathbb{E}D_{1v-1}\ ), \text{ perform adjustment.} \tag{12}$$

Given that the order schedule is determined at the beginning of the planning horizon, we can also consider the demand since the beginning of the planning horizon. This condition expressed in 12 compares the expected demand from period 1 through $v-1$, $\mathbb{E}D_{1v-1}$, with the realized demand $\hat{D}_{1v-1}$. This evaluates the adequacy of the order schedule based on the demand up to the current period.

In addition to assessing the accuracy of our demand expectations based on absolute quantity, it can be informative to consider the fluctuations in the realized demand. A significant difference between the expected and realized variance can be indicative of changes in demand behavior that were not initially anticipated, and thus serve as an indicator for beneficial schedule adjustments. We therefore introduce adjustment condition (13) which is based on the variance of realized demand. This condition monitors whether the realized variance of demand within a replenishment cycle significantly exceeds the variance of the expected demand in the time since the last order period. To ensure the adjustment is only performed in the case where the demand is lower than expected, the realized demand in that time-frame must also be lower then the expected demand.

$$\text{If } (\hat{\sigma}^2_{uv-1} \geq \gamma \cdot \sigma^2_{uv-1} \text{ and } \hat{D}_{uv-1} \leq \mathbb{E}D_{uv-1}), \text{ perform adjustment.} \tag{13}$$

In this equation, $\hat{\sigma}^2_{uv-1}$ represents the variance of the realized demand within the previous replenishment cycle, while σ^2_{uv-1} represents the variance of the demand expected at the start of the planning horizon. This condition provides another dimension to our approach, enabling the algorithm to not only respond to changes in demand magnitude but also adapt to changes in demand variability. In doing so, this can offer more refined control over the ordering process, thereby potentially enhancing the efficiency of the entire supply chain.

For future research, this heuristic can be expanded to include more conditions and possible adjustments. It could be interesting to also impose a condition for when demand is a lot larger then expected. More research could provide another adjustment action, which may be less computationally exhaustive then solving the entire PM model. Another intriguing extension is the incorporation of trend analysis into the adjustment process. The adjustment criteria could be informed by external data sources that predict demand changes, such as media reports or trends in social media or search engine activity. Despite these promising possibilities for expanding the heuristic, the computational experiments in this thesis will be limited to adjustments based on condition (11) due to time and practical constraints.

4.2.3 Numerical example

To illustrate the working of the proposed algorithm, we consider a numerical instance characterized by a lumpy demand pattern with parameters $N = 10$, $h = 1$, $p = 5$, $\alpha = 0.9$, and $\rho = 0.1$. The **expected demand** in this instance is represented by $[7, 17, 16, 5, 2, 297, 8, 19, 4, 5]$. The PM model is initially solved, yielding the replenishment cycles $(0, 5)$, $(5, 6)$, and $(6, 10)$. To evaluate this original solution, we will simulate new demands according to the same distribution. These **realized demand** is given by $[1, 11, 11, 5, 0, 40, 60, 61, 0, 9]$. In order to explore possible improvements, every order period beyond the first (in this case, periods 5 and 6) will be evaluated according to adjustment condition (11). According to this condition, an adjustment is performed if the realized demand in the time-frame since the last order period is at least a factor γ smaller than the expected demand in that period. In this particular instance, we set $\gamma = 0.3$.

The first evaluation is made on order period 5. Comparing expected and realized demands since the last order period (periods 0 through 4) reveals that the realized demand is lower than the expected demand up to period 5. The realized demand in this time-frame is 28, while the expected demand is 47. With $\gamma = 0.3$, the condition for adjustment does not hold and we proceed to evaluate the next order period.

For order period 6 we evaluate the demand since the last evaluated period (period 5). The realized demand is 40, while the expected demand is 297, which is more than a factor 0.3 larger, thus triggering an adjustment. Period 6 is excluded from being an order period, and the PM model is solved again for all periods after period 6. Solving the PM model from period 6 onward yields replenishment cycles $(0, 5)$, $(5, 7)$ and $(7, 10)$.

Again, we asses the adjustment condition for the next order period (period 7), which evaluates the demand since the last evaluated period. In period 6 the realized demand is 60, and the expected demands is 8. The condition for adjustment does not hold, thus we keep order period 7 in the solution. After all order periods are checked, we evaluate the effects of the adjustment via the realized costs and number of adjustments. If we compute the costs according to the realized demands, the adjustment of excluding period 6 has led to a 1.48% cost decrease. This clearly illustrates the benefit of re-evaluating the order-periods and shows that even though an order period is added, the cost can decrease via this adjustment.

5 Results

In this Section, the results are discussed. We first discuss the computational experiment and results results of the PM model. After which the results of the proposed algorithm using semi-static-dynamic uncertainty strategy are presented. All computational experiments are conducted utilizing Java CPLEX on a computing system equipped with a 1.80 GHz Intel Core i7-8565U processor and 16GB of RAM. The code used is shortly described in section A of the Appendix.

5.1 PM model

5.1.1 Computational experiment: PM model

In the interest of performance analysis, we will perform a number of simulations with the PM model with the following parameter settings:

- Holding costs $h = 1$
- Back-order costs $p = 5$
- Setup costs $K = \{225, 900, 2500\}$
- Service level $\alpha = \{0.90, 0.95, 0.99\}$
- Planning horizon $N = \{20, 30, 40\}$

The demand is assumed to be normally distributed with coefficient of variation ρ. We consider the cases of $\rho = \{0.1, 0.2, 0.3\}$. In addition, we consider two demand patterns, $\pi = \{Erratic, Lumpy\}$. For the erratic pattern, the mean demand μ is drawn from a uniform distribution on the interval $[0, 100]$. For the lumpy pattern, the mean demand μ is uniformly drawn from different intervals. With a probability of 20%, the mean demand is drawn on $[0, 420]$, and with a probability of 80% the mean demand is drawn on $[0, 20]$. The lumpy generation has a larger variability and is considered more unpredictable. Ten instances are generated for all possible combinations of these parameters, resulting in 1620 problem instances. We compare the objective value and computation time in seconds by averaging over all problem instances characterized by the same parameter.

5.1.2 Results: PM model

In Table 1, the results of our implementation of the PM model (thesis results) are presented next to the results of the PM model of Tunc et al. (2018). Both models solve the stochastic lot-sizing problem according to the extended formulation of Tunc et al. (2018) as described in Section 4. We compare the results across various parameters, namely the demand pattern, the planning horizon, the co-variance, the fixed costs, and the required service level.

		Thesis results		Tunc (2018)	
Parameters	**Type**	**Objective**	**TIME**	**Objective**	**TIME**
π	*Erratic*	13453,65	0,13	13367,86	1,27
	Lumpy	8584,38	0,18	8591,71	1,33
N	*20*	7233,06	0,08	7277,48	0,19
	30	11122,55	0,14	11063,16	0,84
	40	14701,42	0,25	14598,71	2,87
ρ	*0,1*	9856,06	0,13	9831,21	1,25
	0,2	11022,90	0,15	10984,41	1,31
	0,3	12178,07	0,19	12123,73	1,35
K	*225*	4936,79	0,21	4902,08	1,36
	900	10298,48	0,13	10261,19	1,3
	2.500	17821,76	0,13	17776,08	1,24
α	*0,9*	10474,94	0,18	10378,43	1,48
	0,95	10909,31	0,17	10843,26	1,48
	0,99	11672,78	0,11	11717,66	0,95
Average		11019,01	0,16	10979,78	1,3

Table 1: Comparison of the objective value and computation time in seconds of the replication results versus the original results

The overall findings suggest that both models demonstrate considerable efficiency. On average, our implementation results in an objective that is 0.4% larger than the objective value of Tunc et al. (2018). The running times of our implementation are smaller. These difference could be attributed to varying efficiencies of the CPLEX and Gurobi solvers, differences in processor speed, or inefficiencies in implementation. The relative integrality gap (SGAP), relative optimality gap (EGAP), and the number of nodes are similar for both implementations and can be found in Table 6 of the Appendix.

5.2 Semi-static-dynamic uncertainty strategy

5.2.1 Computational experiment: Semi-static-dynamic uncertainty strategy

In order to examine the effectiveness of the proposed heuristic for the semi-static-dynamic uncertainty strategy, a series of computational experiments is conducted over six instances. For these experiments, the fixed cost (K), the coefficient of variation (ρ), and the service level (α) are held constant at the values of 225, 0.1, and 0.9, respectively. We use both the lumpy and erratic demand distribution patterns to compare our findings across various demand conditions. The planning horizons considered are of lengths $N = \{20, 30, 40\}$.

For each instance, we generate a set of 500 realized demand scenarios with the same distribution. The heuristic is applied to each scenario, resulting in a pair of schedules - one before and one after the adjustments. The adjustment condition used in these experiments is the one described by Equation (11) as presented in Subsection4.2.2. By comparing the objective values obtained from these schedules, we are able to quantify the objective gain brought by the semi-static strategy. Additionally, we keep track of the number of adjustments made during the process, as an excessive number of adjustments may not be practical in real-life scenarios. By analyzing the objective gain and the number of adjustments, we can assess the performance and practicality of the semi-static strategy in solving the stochastic lot-sizing problem under static-dynamic uncertainty. We replicate this process over the range of γ values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, offering insights into how the choice of this parameter influences the overall outcomes of our strategy.

To provide a more comprehensive perspective on the influence of the γ parameter, we supplement our original set of experiments with an additional 250 realized demand scenarios for the values of $\gamma = 0.05, 0.15, 0.25, 0.35, 0.45$. The results from these additional runs are combined with the other experiments to present a graphical view of the effects of the γ parameter.

5.2.2 Results: semi-static-dynamic uncertainty strategy

Taking an overview of the results, on average, our semi-static-dynamic uncertainty strategy prompts adjustments in 89% of the instances, leading to an average objective decrease of 0.066%. The 95% confidence interval for the average objective decrease is given by $[0.062, 0.070]$. The average number of adjustments made during the process is 1.79. Noting that the objective decrease when no adjustments are made equals zero, the average objective decrease in case of adjustments equal to 0.176%.

γ	**Adjusted**	**Objective increase**	**Confidence interval**	**Adjustments**
0.1	22%	0.001%	[0.000, 0.002]	0.09
0.2	49%	0.011%	[0.007, 0.014]	0.34
0.3	63%	0.028%	[0.022, 0.034]	0.75
0.4	76%	0.067%	[0.058, 0.076]	1.97
0.5	87%	0.223%	[2.208, 2.238]	5.78
Average	89%	0.066%	[0.062, 0.070]	1.79

Table 2: Solution statistics for different values of γ

Table 2 shows how the γ value impacts the performance of the semi-static-dynamic uncertainty strategy. As γ increases, the number of instances that trigger the adjustment condition also rises. Specifically, when $\gamma = 0.1$, only 22% of instances called for an adjustment. However, as γ increases to 0.5, 87% of instances were adjusted, showing a clear positive correlation between γ and the frequency of adjustments.

Additionally, Table 2 shows the relationship between γ and the objective function decrease. For larger γ values, more adjustments are made, leading to a higher objective decrease. The lower and upper bounds of the objective decrease were also noted, providing a 95% confidence interval for the performance of the strategy. With higher γ values, both the lower and upper bounds increase, signifying a broader spread in potential outcomes. The number of average adjustments made per run remains below one for $\gamma = 0.1$ to $\gamma = 0.3$ and it shows a considerable jump to almost 2 and more than 5 for γ values of 0.4 and 0.5, respectively. Given that an excessive number of adjustments may not be practical in real-world supply chain scenarios, this finding accentuates the importance of an optimal choice for γ.

N	**Adjusted**	**Objective decrease**	**Adjustments**
20	31%	0.095%	0.90
30	34%	0.030%	1.21
40	47%	0.073%	3.26
Average	37%	0.066%	1.79

Table 3: Solution statistics for different planning horizons (N)

Table 3 shows the effects of the planning horizon. As expected, a longer planning horizon, which encompassing more periods, increases the likelihood of adjustments. Specifically, adjustments occurred in 31%, 34%, and 47% of instances for planning horizons of 20, 30, and 40 periods respectively. On examining the objective function decreases, it is noticeable that the decrease is not linear with the increase in the planning horizon size. While the decrease is 0.095% for a planning horizon of 20 periods, it drops to 0.030% for 30 periods and then rises slightly to 0.073% for 40 periods. These variations across differing planning horizons could be due to the amount of uncertainties across a greater number of periods. With a longer horizon, the model has more opportunities to encounter and react to unforeseen variations in demand, which might not necessarily align with an objective decrease. With regard to the number of adjustments, there's a clear upward trend as we extend the planning horizon. For $N = 20$, the average number of adjustments stands at 0.90, which then jumps to 1.21 for $N = 30$ and further to 3.26 for $N = 40$. A longer horizon may invoke more adjustments, but not necessarily result in a proportionate decrease in the objective function.

Demand pattern	**Adjusted**	**Objective decrease**	**Adjustments**
Lumpy	45%	0.044%	1.05
Erratic	30%	0.088%	2.53
Average	37%	0.066%	1.79

Table 4: Solution statistics for different demand patterns

The comparison between the lumpy and erratic distribution is shown in Table 4. Despite adjustments being performed in 45% of the instances, the lumpy distribution, on average, triggers fewer adjustments. The erratic distribution initiates more adjustments, albeit only in 30% of instances. Interestingly, the average objective decrease is larger for the erratic pattern, despite fewer instances of adjustments. These differences can be attributed to the characteristics of these demand patterns. The lumpy pattern, marked by a higher degree of variability, leads to a larger number of adjustments, while the erratic pattern, with its relatively lower variability, results in fewer instances where adjustments are triggered.

To see how the different distributions react to different values of γ, we compare them in Table 5. It is clear that the different distributions react differently to the parameter settings of γ. For the lumpy pattern, the number of adjusted periods incrementally increases with rising γ. However, the erratic pattern exhibits a more dramatic reaction, with a sharp increase in the amount of adjustments when γ shifts from 0.4 to 0.5. It is interesting to see that for $\gamma = 0.4$, the objective decrease is nearly identical, while the number of adjustments is 1.48 for the lumpy distribution and 2.47 for the erratic distribution. This suggests that our adjustment algorithm may operate with greater efficiency in scenarios characterized by a lumpy distribution, achieving

γ	Demand pattern	Objective decrease	Adjustments
0.1	Lumpy	0.002%	0.17
	Erratic	0.000%	0.02
0.2	Lumpy	0.021%	0.58
	Erratic	0.000%	0.10
0.3	Lumpy	0.047%	1.00
	Erratic	0.009%	0.50
0.4	Lumpy	0.067%	1.48
	Erratic	0.068%	2.47
0.5	Lumpy	0.083%	1.99
	Erratic	0.363%	9.58

Table 5: Solution statistics for the demand patterns across different values of γ

similar decreases in objective function with fewer adjustments. These findings underscore that the selection of γ must be finely tuned to the specific demand pattern under consideration to optimize the performance of the semi-static-dynamic uncertainty strategy.

The comparative analysis of different demand patterns under the influence of varied γ values is visually represented in Figures 1 and 2. These figures show the average objective decrease and number of adjustments for the lumpy and erratic distribution, respectively.

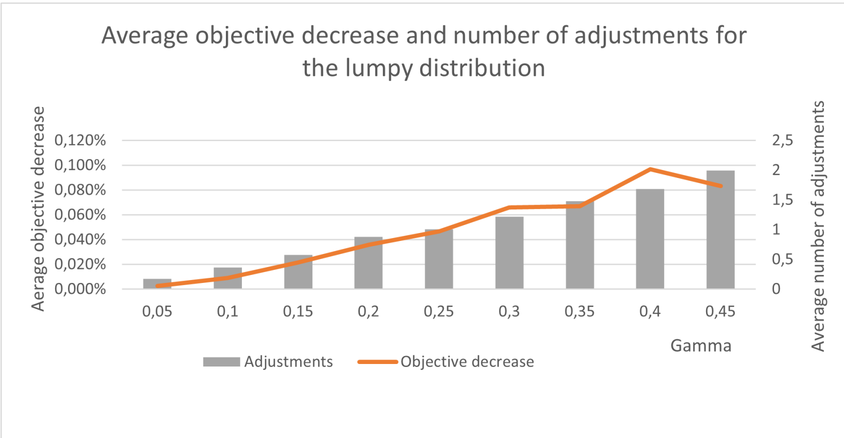

Figure 1: Average objective decrease and number of adjustments for the lumpy distribution

As shown in Figure 1, the response of the lumpy demand pattern to an increase in γ is rather consistent. A steady increment in the number of adjusted periods and objective decrease is observed with increasing γ values. This suggests an improved cost efficiency with a moderate rise in operational adjustments. This implies that, with a lumpy demand pattern, the semi-static-dynamic uncertainty strategy can yield consistent improvements in cost efficiency without demanding a drastic increase in operational flexibility.

In contrast, the erratic demand pattern presents a different scenario, as shown in Figure 2. The Figure shows a sharp incline in the number of adjustments and objective decrease as γ increases beyond 0.35. Hence, for erratic demand patterns, while the semi-static-dynamic strategy can produce significant cost savings, it comes with the trade-off of requiring a considerably higher level of adaptability in operations.

In conclusion, these findings highlight the nuanced relationship between the γ parameter

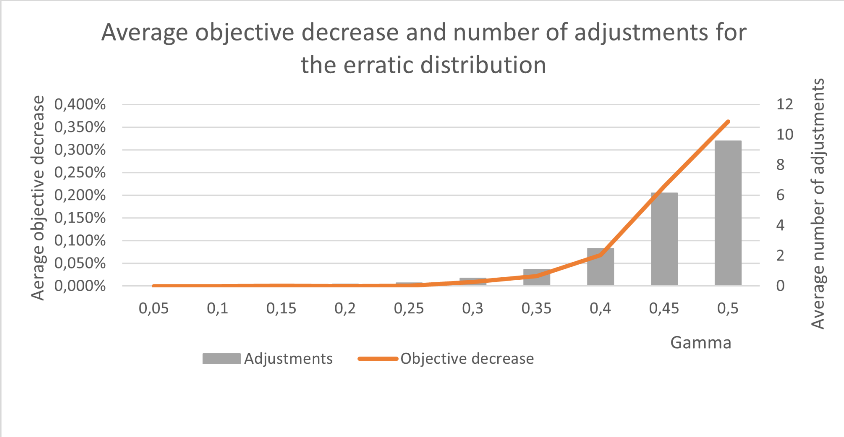

Figure 2: Average objective decrease and number of adjustments for the erratic distribution

and demand patterns in optimizing the semi-static-dynamic uncertainty strategy. Understanding these interactions can enable better tuning of the strategy to improve cost efficiency and manage the degree of required operational adjustments. It shows the need to customize the γ parameter based on the characteristics of the demand pattern, optimizing the balance between cost efficiency and operational flexibility.

6 Conclusion

In this paper, we have addressed the stochastic lot-sizing problem under static-dynamic uncertainty strategy and proposed the semi-static-dynamic uncertainty strategy as an innovative approach. First, we replicated the results of the MIP model as proposed by Tunc et al. (2018). Our implementation demonstrated comparable efficiency and provided objective values that were on average 0.4% larger than the original results. This difference could be due to variations in solver efficiencies, processor speeds, or implementation details.

Second, we introduced the semi-static-dynamic uncertainty strategy, which blends the static-dynamic and dynamic uncertainty strategies to enhance adaptability and induce cost savings. The proposed heuristic algorithm allows for adjustments to the initial replenishment schedule based on realized demand information, improving the responsiveness to demand fluctuations. By evaluating one adjustment condition, we demonstrated the potential benefits of the semi-static-dynamic strategy. The algorithm follows a heuristic approach, which involves solving the problem initially using the static-dynamic strategy. Then all order periods are evaluated based on an adjustment condition. If the realized demand data calls for an adjustment, the order period will be excluded from the solution and the PM model will be solved again for all future periods. On average we found that when adjustments are made, the heuristic induces a small cost saving of 0.176%.

The adjustment condition's sensitivity is determined by the parameter γ. A larger value imposes a stricter condition, thus inflicting more adjustments. The greater the amount of adjustments, the greater flexibility that is expected of the operation. It is thus necessary to strike a balance between cost-reduction and practical feasibility using the parameter value γ. Our analysis was made on two distributions, where the distribution with higher volatility induced better cost savings while requiring fewer adjustments to the original schedule.

Our research primarily centers around the heuristic evaluation based on an adjustment condition that compares the realized demand since the last order period. This simple adjustment condition has proven to induce small yet consistent cost savings. However, the heuristic adjustment condition and its resulting impact on the stochastic lot-sizing problem presents many opportunities for exploration. Future research could benefit from investigating and incorporating other factors as external predictors for fluctuating demand. These predictors could potentially include factors such as market trends, seasonality, volatility, and economic indicators. By expanding the range of adjustment conditions, the heuristic can become more adaptable and responsive to real-world situations. In essence, further research into the development of varied adjustment conditions not only enhances the flexibility and adaptability of our proposed heuristic but also presents a pathway towards a more nuanced understanding of the practical use of the stochastic lot-sizing problem.

In conclusion, the introduction of the semi-static-dynamic uncertainty strategy offers an innovative approach that improves upon the traditional static-dynamic strategy by incorporating adjustments during the planning horizon. The results highlight the potential of the semi-static-dynamic strategy in achieving cost savings and better adaptation to demand fluctuations. Future research can focus on testing additional adjustment conditions and evaluating the performance of the semi-static-dynamic strategy under different parameter settings. The heuristic presented in this thesis holds promise for advancing the field of stochastic lot-sizing solutions and its practical applications.

References

Bookbinder, J. H. & Tan, J. (1988). Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, *34*(9), 1096–1108.

Rossi, R., Kilic, O. & Tarim, S. (2015). Piecewise linear approximations for the static-dynamic uncertainty strategy in stochastic lot-sizing. *Omega*, *50*, 126–140.

Rossi, R., Tarim, S., Hnich, B. & Prestwich, S. (2011). A state space augmentation algorithm for the replenishment cycle inventory policy. *International Journal of Production Economics*, *133*(1), 377–384.

Rossi, R., Tarim, S., Prestwich, S. & Hnich, B. (2014). Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, *231*, 489–502.

Tarim, S., Dogru, M., Ozen, U. & Rossi, R. (2011). An efficient computational method for a stochastic dynamic lot-sizing problem under service-level constraints. *European Journal of Operational Research*, *215*(3), 563–571.

Tarim, S. & Kingsman, B. (2004). The stochastic dynamic production/inventory lot-sizing problem with service-level constraints. *International Journal of Production Economics*, *88*(1), 105–119.

Tarim, S. & Kingsman, B. (2006). Modelling and computing (rn,sn) policies for inventory systems with non-stationary stochastic demand. *European Journal of Operational Research*, *174*(1), 581–599.

Tunc, H., Kilic, O., Tarim, S. & Eksioglu, B. (2014). A reformulation for the stochastic lot sizing problem with service-level constraints. *Operations Research Letters*, *42*(2), 161–165.

Tunc, H., Kilic, O. A., Tarim, S. A. & Rossi, R. (2018). An extended mixed-integer programming formulation and dynamic cut generation approach for the stochastic lot-sizing problem. *INFORMS Journal on Computing*, *30*(3), 492–506.

Özen, U., Doğru, M. & Tarim, S. (2012). Static-dynamic uncertainty strategy for a single-item stochastic inventory control problem. *Omega*, *40*(3), 348–357.

A Programming code

In this Section, we will provide a description of the code used to obtain the results found in Section 5.

A.1 PM model

In order to solve the PM model, we import the parameter values from the file provided by Tunc et al. (2018). We call the *solvePM()* method with the instance parameters and two lists which are used for the semi-static-dynamic algorithm. To solve the static-dynamic uncertainty strategy lot sizing problem, empty lists must be inputed here.

The *solvePM* method solves a cplex model with the objective and constraints as explained in Section 4. The linearization of the loss function is done by the *LinearApproximation* class, which provides the values of a and b for a standardized linearization with 11 segments. The *InverseCDF* class computes the value of the inverse cumulative distribution function of the random demand, which is used in the service level constraint.

After solving the MIP model, the results will be exported to an excel file using the *ExcelReader* class. All importing and exporting is done via the *ExcelReader* class. It is of importance to change the location of your excel file when calling this method.

A.2 Semi-static-dynamic uncertainty strategy

For the semi-static dynamic algorithm, we will use the same *solvePM* method. To find the initial solution, we use the same inputs as described above. The realized demand is generated by the *DemandGenerator* class, which generates random demand according to the demand pattern, coefficient of variation and the size of the planning horizon.

The class *RealizedCostCalculator* computes the realized costs of a solution using the realized demand. We take note of the original realized costs, to compare it with the realized costs after adjustments. After this, the algorithm is started by the *agorithm* method. The adjustment condition will be checked for all order periods until the condition is found. If an order period is found where the adjustment condition holds, we will go back to solving the *solvePM* method with the adjusted lists for $forbiddenList$ and $setList$. These lists contain the periods we have excluded or have already verified. In the case that *solvePM* finds no optimal solution after excluding a period, the period will be added back into the solution.

When all order periods are checked and all imperative adjustments are made, we compute the adjusted realized costs. This program is solved for six instances, with multiple of γ for a certain number of runs. After all instances are ran and their results are saved, the results are exported to excel. We save the objective gain, the number of forbidden periods, and variable parameters (N, $gamma$, $demandPattern$).

		Thesis results					Tunc (2018)				
Parameters	**Type**	**Objective**	**S-GAP**	**E-GAP**	**TIME**	**NODES**	**Objective**	**S-GAP**	**E-GAP**	**TIME**	**NODES**
π	*Erratic*	13453,65	9,4E-08	9,39E-08	0,13	0	13367,86	0	0	1,27	0
	Lumpy	8584,38	2,5E-07	2,45E-07	0,18	0	8591,71	0,24	0	1,33	0,24
N	*20*	7233,06	4,2E-07	4,22E-07	0,08	0	7277,48	0,05	0	0,19	0,01
	30	11122,55	2,8E-08	2,77E-08	0,14	0	11063,16	0,17	0	0,84	0,15
	40	14701,42	5,9E-08	5,89E-08	0,25	0	14598,71	0,14	0	2,87	0,21
ρ	*0,1*	9856,06	1,6E-16	0,00E+00	0,13	0	9831,21	0,01	0	1,25	0
	0,2	11022,90	2,8E-08	2,77E-08	0,15	0	10984,41	0,11	0	1,31	0,11
	0,3	12178,07	4,8E-07	4,81E-07	0,19	0	12123,73	0,23	0	1,35	0,25
K	*225*	4936,79	4,8E-07	4,81E-07	0,21	0	4902,08	0,28	0	1,36	0,25
	900	10298,48	2,8E-08	2,77E-08	0,13	0	10261,19	0,08	0	1,3	0,11
	2.500	17821,76	6,6E-17	0,00E+00	0,13	0	17776,08	0	0	1,24	0
α	*0,9*	10474,94	4,2E-07	4,22E-07	0,18	0	10378,43	0,06	0	1,48	0,04
	0,95	10909,31	1,9E-16	0,00E+00	0,17	0	10843,26	0,11	0	1,48	0,11
	0,99	11672,78	8,7E-08	8,66E-08	0,11	0	11717,66	0,2	0	0,95	0,22
Average		11019,01	1,7E-07	1,70E-07	0,16	0	10979,78	0,12	0	1,3	0,12

Table 6: Comparison of the replication results versus the original results for the PM model with α service level constraint