Erasmus University Rotterdam

Master Thesis
Econometrics and Management Science

Sustainability in stochastic lot-sizing with coordinated shipments

Author: Roberts Ūpis *(615421)*

Supervisor: dr. Wilco van den Heuvel

Second assessor: Prof dr. Albert Wagelmans

20th October 2023

ERASMUS UNIVERSITEIT ROTTERDAM
ERASMUS SCHOOL OF ECONOMICS

Abstract

This paper considers the application of sustainability in a multi-item stochastic lot-sizing problem by coordinating shipments using augmented ϵ-constraint with bi-objective modeling. The stochastic demand uncertainty is modeled using static, static-dynamic, and static receding horizon approaches with extensions for capacity limitation and aggregated service levels. Additionally, a variable neighborhood descent meta-heuristic is applied to the static-dynamic model. The results show that a significant reduction of periods with shipments can be achieved with a slight increase in total cost in all methods. This provides a cost-efficient way to reduce carbon footprint in transportation. Furthermore, a performed sensitivity analysis shows that the results are consistent with changes in the problem scale and parameters.

Keywords: stochastic lot-sizing, shipment coordination, sustainability, static, static-dynamic, metaheuristic

Contents

1 Introduction

Climate change poses a wide range of risks of economic and societal damages (Diaz & Moore, 2017). With the United Nations proposing the 2030 Agenda for Sustainable Development (United Nations, 2015), EU countries have progressed towards the set sustainability goals (Carrillo, 2022). However, transportation is still a significant source of pollution with logistic activities accounting for 5.5% of the global greenhouse gas emissions, most of which arise from freight transportation (McKinnon et al., 2015). Freight transportation still largely depends on fossil fuels, which significantly contribute to emissions (Zhang et al., 2022). Furthermore, Rizet et al. (2014) found that transportation CO2 emissions can be reduced by decreasing shipment frequency. Therefore, it is vital to consider sustainable approaches in supply chain management.

Production planning has shifted from a focus on pure cost minimization to considering the environmental impact of production models (Khaled et al., 2022). Dynamic lot-sizing problems are commonly used in supply chain modeling in the context of inventory planning or production scheduling in industrial environments, i.e., to model shipments from a supplier to a retailer. In such cases, due to varying product life cycles, the demand is stochastic and non-stationary (Tunc et al., 2014). Therefore, achieving sustainability requires consideration of both the overall cost minimization and the environmental impact. Usually, the objective of lot-sizing is cost minimization while maintaining a predetermined non-stockout probability. However, in response to the global climate crisis, the green transition, and corporate sustainability goals, it is necessary to extend supply chain models to account for the environmental impact. Often, lot-sizing optimizes order plans on a per-item basis leading to many small shipments, which leads to an increase in overall emissions.

This thesis introduces an extension of lot-sizing with shipment coordination. This is modeled using a bi-objective approach to empirically assess the cost-benefit trade-offs between minimizing cost and the environmental impact. To model the demand uncertainty, this thesis first proposes a model using a static approach where the decisions of the amount and the timing of production are made for the entire planning period.

Secondly, a static-dynamic approach where the order periods are decided in advance but the order quantities can be adjusted after the demand is realized (Bookbinder & Tan, 1988). Additionally, a receding horizon approach is presented in which the planning is implemented for a limited number of periods and re-planned after demand has been realized (Tavaghof-Gigloo & Minner, 2021). This approach is often used to model empirical implementations of dynamic lot-sizing (Tavaghof-Gigloo & Minner, 2021).

To further account for empirical application when the logistics system is limited by transportation capacity or a worker shortage (Wang et al., 2022), the model is extended with capacity constraints. Finally, in cases with a large variety of products, using an aggregate service level across all products, in addition to a per-product service level, can lead to cost reductions (Sereshti et al., 2021). Therefore, the aggregate service approach is introduced to the model to further reduce the total cost and the number of emissions.

The structure of this thesis is as follows: First, the literature review and the motivation of the thesis are presented. Second, the static and static dynamic versions of the stochastic lot-sizing model are introduced. Third, additional model extensions and a metaheuristic method are discussed. Next, the analysis of the data set, model evaluation, and sensitivity analysis are performed. Lastly, the conclusion is presented.

2 Theoretical Framework

Since the initial lot-sizing sizing model proposed by Harris (1913), such lot-sizing models have been widely researched (Aloulou et al., 2014; Glock et al., 2014). To deal with uncertainty, researchers have applied dynamic and stochastic methods to lot-sizing models (Aloulou et al., 2014; Glock et al., 2014).

Bookbinder and Tan (1988) examined three ways to model lot-sizing problems with stochastic demand uncertainty. Firstly, a static approach where the decisions of amount and timing are made for the entire planning period. Secondly, a static-dynamic approach where the order periods are decided in advance but the order quantities can be adjusted after the demand is realized. Lastly, a dynamic approach where the production amount and timing can be readjusted at every period.

Bookbinder and Tan (1988) present an initial single-item static model with a cycle α service level, that ensures that the non-stockout probability at the end of any period does not become negative. Furthermore, Helber et al. (2013) introduce an additional β^ccycle fill rate service level that ensures the probability that demand in a period is filled from the current inventory. Similarly, Tempelmeier and Herpers (2011) extend the model for multiple items using a δ service level that limits the ratio of the expected and the maximum backlog. Additionally, Tempelmeier (2011) and Tempelmeier and Herpers (2010) introduce additional heuristic methods to solve the problem.

Furthermore, Sereshti et al. (2021) extend on previous static models by introducing multi-item static models across various service levels, including aggregate service levels. They additionally introduce a receding horizon approach, also called the rolling horizon approach, which involves re-evaluating the model after each period based on realized demand. Their approach also includes an additional penalty parameter to penalize any service-level violations. Moreover, Tavaghof-Gigloo and Minner (2021) show that replanning opportunities improve performance when there is no capacity limit, and Forel and Grunow (2022) integrate receding horizon with demand forecasting.

Comparatively, for the static-dynamic approach, Bookbinder and Tan (1988) present a single-item static-dynamic model with a cycle α service level. In the model, the order schedule is determined and afterwards the order-up-to level is calculated. Tarim and Kingsman (2004) further introduce a mixed integer programming (MIP) single-item model formulation that simultaneously determines both the order-up-to level and the order schedule. Tunc et al. (2014) expand on the Tarim and Kingsman (2004) formulation by translating the order schedule into separate replenishment cycles (the time between two production periods), which can efficiently solve large-scale problems to optimality. Furthermore, Tempelmeier (2007) introduces a model with a fill rate β and a cycle α service level. In addition, Rossi et al. (2015) extend the Tarim and Kingsman (2004) model by incorporating various service measures using piece-wise upper and lower bounds of the first-order loss function. Subsequently, Tunc et al. (2018) improve on the time efficiency of the previous formulations by introducing a mixed integer programming model using a dynamic cut generation approach. Özen et al. (2012) introduce two heuristics using dynamic programming for the single-item static-dynamic problem. Randa et al. (2019) further extend the dynamic programming methods and introduce improvement-based local search heuristics. These heuristics use merging, splitting, and shifting production cycles in different combinations and order of operations. They show that a combination of these operations achieves an average optimality gap of 0.1%.

In coordinated lot-sizing problems, an additional setup cost is incurred when any product is produced (Gao et al., 2008; Robinson et al., 2009). Robinson et al. (2009) show that the problem is NP-complete and even deterministic demand problems are difficult to solve on a large scale.

Capacitated lot-sizing problems have been widely studied as a separate category of models (Quadt & Kuhn, 2008). Capacity constraints are typically added to models with static strategy (Sereshti et al., 2021). Tavaghof-Gigloo and Minner (2021) used soft service-level constraints to guarantee feasibility under capacity constraints.

In recent years, studies in inventory management have shifted towards finding more sustainable ways of inventory management. Commonly, carbon dioxide (CO2) emissions are used to measure emission reduction. Examples of such applications include the economic order quantity (EOQ) models (Hua et al., 2011), economic production quantity (EPQ) models (Sepehri & Gholamian, 2022), and integrated production scheduling models (Yağmur & Kesen, 2023). Kadziński et al. (2017) show that in supply chain modeling CO2 emissions can be significantly reduced with a slight increase in total cost. Heck and Schmidt (2010) introduce a lot-sizing model that considers ecological factors: power usage, CO2 emissions, and water consumption. Furthermore, Retel Helmrich et al. (2015) present a model with a CO2 emission constraint. Similarly, Vaez et al. (2019) propose a bi-objective lot-sizing model with total cost minimization and CO2 emission minimization. Moreover, Liu (2016) uses lot-sizing models in a renewable energy generation context while reducing CO2 emissions. The models show that lot-sizing models can be effectively used for both cost minimization and emission reduction.

Subsequently, when considering lot-sizing models with bi-objective functions, Romeijn et al. (2014) and van den Heuvel et al. (2012) showed that finding certain formulations of the problem can be solved in polynomial time, but that in general, the problem is NP-hard. Earlier solution methods for bi-objective functions use the weighted-sum approach, which determines Pareto optimal solution by systematically changing the weights between the objective functions (Kim & De Weck, 2005). However, this method often leads to uneven distribution of points along the Pareto front (Das & Dennis, 1998). Additionally, Mavrotas (2009) proposes the augmented ϵ-constraint method that efficiently produces Pareto optimal solutions. Mavrotas and Florios (2013) and Nikas et al. (2022) provide further improvement for the method. Furthermore, Vaez et al. (2019) apply the augmented ϵ-constraint method to a bi-objective lot-sizing problem. Comparatively, other studies have applied the lexicographic weighted Tchebycheff method (Liu, 2016) or proposed meta-heuristic approaches to solve bi-objective models (Yağmur & Kesen, 2023).

This study expands on the existing literature in several ways. Firstly, it extends the application of lot-sizing models to sustainability. Secondly, it employs a novel approach to shipment coordination through bi-objective modeling. Furthermore, this thesis extends the formulations to static, static-dynamic, and receding horizon approaches. Correspondingly, the Tunc et al. (2014) static-dynamic formulation is extended to include multiple items. This thesis further provides empirical applications by introducing capacity constraints and the comparison between aggregate service levels and individual service levels. Lastly, this thesis introduces a metaheuristic approach to the static-dynamic approach.

3 Problem formulation

The focus of this analysis is on the multi-item stochastic lot-sizing problem. A finite planning horizon with T periods and N items is considered. In each period $t \in T$ in the planning horizon, each item $i \in N$ occurs a stochastic and independently distributed demand d_{it} with a known distribution function having a mean μ_{it} and a standard deviation σ_{it}. Further, orders incur a fixed set-up cost f_i and each unit in inventory incurs a holding cost h_i. In the capacitated version of the problem, no more units than C_t

can be produced. Unused units are carried from one period to the next, and any demand that exceeds inventory is back-ordered. Additionally, it is required to meet a set service level α_i for all items. See Table 1 for an overview of used parameters, random variables, and variables.

Table 1: Static model parameters, random variables, and variables

Sets	
T	Set of the time periods in the planning horizon, indexed by t
N	Set of all items, indexed by i
Parameters	
f_i	Set up costs for item i
h_i	Holding costs for item i
$I_{i,0}$	Inventory level for item i at period 0
α_i	Minimum service level for item i
P	Emission setup cost in each period
M, M_2	Sufficiently large numbers
μ	Sufficiently small number
Random variables	
$D_{i,t}$	Demand for item i in period t
$F_{i,t}$	The cumulative distribution of the random demand from period 1 to period t for item i
$F_{i,t}^{-1}(\alpha_i)$	The minimum value of cumulative demand δ for item i from period 1 to period t for which $P(\sum_{j=1}^{t}(D_{i,j}) \leq \delta) \geq \alpha_i$
$I_{i,t}$	Inventory level for item i in period t

Continued on next page

Table 1: Static model parameters, random variables, and variables (Continued)

Variables	
$x_{i,t}$	Production amount for product i in period t
$y_{i,t}$	Binary variables that equals 1 if there is production for item i in period t, 0 otherwise
z_t	Binary variable that equals 1 if there is any production in period t
s	Slack variable

3.1 Static demand model

The static demand model is based on Tempelmeier (2007) and is extended for multiple items similar to Sereshti et al. (2021).

$$Z_1 = \min \sum_{i \in N} \sum_{t \in T} (f_i y_{it} + h_i E[I_{it}]) \quad (1)$$

$$\text{s.t.} \quad I_{i,t} = I_{i,0} + \sum_{j=1}^{t} (x_{i,j} - E[D_{i,j}]) \qquad \forall i \in N, \forall t \in T \quad (2)$$

$$x_{i,t} \leq M y_{i,t} \qquad \forall i \in N, \forall t \in T \quad (3)$$

$$I_{i,0} + \sum_{j=1}^{t} x_{i,j} \geq F_{i,t}^{-1}(\alpha_i) \qquad \forall i \in N, \forall t \in T \quad (4)$$

$$y_{i,t} \in \{0, 1\} \qquad \forall i \in N, \forall t \in T \quad (5)$$

$$x_{i,t} \geq 0 \quad I_{i,t} \geq 0 \quad \forall i \in N, \forall t \in T \quad (6)$$

In the model, the objective (1) minimizes the total setup and holding costs. Constraint (2) is the balance constraint of the expected inventory. Constraint (3) ensures that the set-up costs are accounted for during production. Here the number M needs to be large enough to enforce the constraint $M > \max_i(F_{0,T}^{-1}(\alpha))$. Constraint (4) is the individual service level constraint that the sum of the initial inventory level and the production quantity for item i until period t is larger or equal to the cumulative demand required to ensure minimum service level. Lastly, constraints (5) and (6) show the domain of the variables.

Additionally, the net expected inventory is assumed to be positive as the amount of negative inventory is negligible when considering large α service levels Tempelmeier (2007).

Next, we introduce a second objective function (7) that accounts for any item's emissions arising from set-up costs. Constraints (8) and (9) account for an emission setup cost when production occurs in a period t. In a shipping context, any period with shipments incurs a penalty, thus minimizing the number of shipments and reducing the total emission amount. The number M needs to be large enough to enforce the constraint when production occurs for all items $M_2 > I$.

$$Z_2 = \min \sum_{t \in T} P z_t \tag{7}$$

$$\text{s.t.} \quad \sum_{i \in N} y_{i,t} \leq M_2 z_t \qquad \forall t \in T \tag{8}$$

$$\sum_{i \in N} y_{i,t} \geq z_t \qquad \forall t \in T \tag{9}$$

Continuing, the model is reformulated using the augmented ϵ-constraint method (Mavrotas, 2009; Sadjadi et al., 2014; Vaez et al., 2019). Here objective (1) is reformulated to objective (10) and objective (7) is further reformulated as the constraint (11). Using the slack variable s only efficient solutions are produced (Vaez et al., 2019). The parameter μ is a sufficiently small number, in the range from 10^{-6} to 10^{-3} depending on the relative size of Z_1 and Z_2 (Mavrotas, 2009). The ϵ is determined in a similar manner to Vaez et al. (2019). In constraint (11), a set of evenly distributed ϵ values are taken in the range between the minimum and the maximum value of Z_2. The maximum value of Z_2 is obtained by, firstly, solving the model with the objective function (1), constraints (2, 3, 4, 5, 6), and the additional constraints (8) and (9). Then the maximum value of Z_2 is determined by calculating the value of the objective function (7) with the obtained optimal decision variables. Next, the minimum value of Z_2 is determined by minimizing the objective (7) with the constraints (2, 3, 4, 5, 6, 8, 9).

$$\min(Z_1 + \mu s) \tag{10}$$

$$\text{s.t.} \quad Z_2 = \epsilon - s \tag{11}$$

3.2 Static-dynamic demand model

The static-dynamic formulation is based on the MIP formulation by Tunc et al. (2014). They reformulate the problem to find the lowest-cost replenishment cycles i.e., the time between two production periods, in the order schedule. This is further extended to incorporate multiple items. For each item i lowest-cost replenishment cycles are determined. See Table 2 for used parameters and variables.

Table 2

Additional Static-dynamic model parameters, random variables, and variables

Sets	
$C_{t,j}$	Set of all replenishment cycles where t and j are successive replenishment periods
Parameters	
$c_{i,t,j}$	The lower bound of the cycle costs for an item i in a cycle from period t to period j
Random variables	
$G_{i,t,j}$	The cumulative distribution of the random demand from period t to period j for item i
Variables	
$X_{i,t,j}$	Binary variable that equals 1 if periods t and j are successive replenishment ($t < j$) periods for item i, 0 otherwise
$S_{i,t,j}$	The order-up-to level for item i in period t if t and j are successive replenishment periods, 0 otherwise
Z_t	Binary variable that equals 1 if any replenishment period is in period t

The static-dynamic model formulation is given below:

$$Z_1 = \min \sum_{i\in N}\sum_{t=1}^{T}\sum_{j=i+1}^{T+1}(c_{i,t,j}X_{i,t,j} + h_i(j-t)(S_{i,t,j} - G^{-1}_{D_{i,t},\ldots D_{i,j-1}}(\alpha_i)X_{i,t,j})) \quad (12)$$

$$\text{s.t.} \sum_{j=t+1}^{T+1} X_{i,t,j} - \sum_{j=1}^{t-1} X_{i,j,t} = 0 \qquad \forall i \in N, t \in [2,T] \quad (13)$$

$$\sum_{j=2}^{T+1} X_{i,1,j} = 1 \qquad \forall i \in N \quad (14)$$

$$\sum_{t=1}^{T} X_{i,t,T+1} = 1 \qquad \forall i \in N \quad (15)$$

$$S_{i,t,j} \le MX_{i,t,j} \qquad \forall i \in N, t \in [1,T], j \in [t+1, T+1] \quad (16)$$

$$S_{i,t,j} - G^{-1}_{i,t,j-1}(\alpha_i)X_{i,t,j} \ge 0 \qquad \forall i \in N, t \in [1,T], j \in [t+1, T+1] \quad (17)$$

$$\sum_{j=1}^{t-1} S_{i,j,t} - \sum_{j=1}^{t-1}(X_{i,j,t}\sum_{k=j}^{t-1} E[D_k]) \le \sum_{j=t+1}^{T+1} S_{i,t,j} \qquad \forall i \in N, t \in [2,T] \quad (18)$$

$$S_{i,t,j} \ge 0, X_{i,t,j} \in \{0,1\} \qquad \forall i \in N, t \in [1,T], j \in [t+1, T+1] \quad (19)$$

Where a given replenishment cycle is from period j to $k-1$ and the lower bound of the cycle cost $c_{i,t,j}$ can be calculated as:

$$c_{i,t,j} = f_i + h_i \sum_{k=t}^{j-1}(G^{-1}_{i,t,j-1}(\alpha_i) - \sum_{l=t}^{k} E[D_{i,l}]) \quad \forall i \in N,\ t \in [1,T],\ j \in [t+1, T+1] \quad (20)$$

The minimum cycle cost can be pre-calculated, which substantially reduces model run time (Tunc et al., 2014). The objective function (12) minimizes the total costs over all possible replenishment cycles for all items. Constraint (13) ensures that a new consecutive cycle starts after the first one ends for each item while constraints (14) and (15) ensure that there is a starting and an end cycle for each item. Continuing, constraint (16) ensures that order-up-to levels are only in the replenishment periods and constraint (17) enforces the minimum order-up-to level. Next, constraint (18) ensures that the consecutive cycle order-up-to level is larger than the expected inventory level at the end of the previous cycle. Lastly, constraint (19) shows the domain of the variables.

Similar to the static model we introduce a second objective function (21) that accounts for the emissions arising from set-up costs for any item. Constraints (22) and (23) account for an emission setup cost when producing a product. The objective is then reformulated in the same way.

$$
\begin{aligned}
Z_2 \;&=\; \min \sum_{t \in T} PZ_t && && (21)\\
\text{s.t.} \quad & \sum_{i \in N} \sum_{j=t+1}^{T+1} (X_{i,t,j}) \leq M_2 Z_t && \forall t \in T && (22)\\
& \sum_{i \in N} \sum_{j=t+1}^{T+1} (X_{i,t,j}) \geq Z_t && \forall t \in T && (23)
\end{aligned}
$$

3.3 Receding horizon approach

In the static model, the production and setup periods are kept unchanged over the planning horizon. However, this can reduce the responsiveness of the system and introduce additional costs (Sereshti et al., 2021). The receding horizon approach accounts for the realized demand by reevaluating the model after each period. This gives the static model similar functionality to the static-dynamic model. However, in contrast to the static and static-dynamic model, the production periods in the receding horizon approach are not fixed and can be changed after the initial schedule. This resembles the dynamic strategy described in Bookbinder and Tan (1988) where both the production timing and amounts are evaluated in every period. Furthermore, the receding horizon strategy can be applied to both static and dynamic models. However, based on computational results, the receding horizon with the static uncertainty model is the best alternative for the dynamic strategy (Dural-Selcuk et al., 2020).

In the receding horizon approach, the static model is initially solved for the entire planning horizon from period 1 to period T using the augmented ϵ-constraint method. The solution with the lowest total cost, i.e. the sum of set-up costs, holding costs, and the emissions set-up cost (24) is fixed for the further period. The inventory of the following period is calculated with (25)

$$\min \sum_{i \in N} \sum_{t \in T} (f_i y_{it} + h_i E[I_{it}]) + \sum_{t \in T} P z_t \tag{24}$$

$$I_{i,t} = I_{i,t-1} + x_{i,t} - RD_{i,t} \tag{25}$$

where $RD_{i,t}$ is the realized demand for item i at time t. The procedure is repeated for each further period until the end of the planning horizon.

3.4 Model Extensions

To further extend the model, two additions are proposed. First, a production capacity constraint is introduced, and second, aggregate service levels are considered.

3.4.1 Capacity constraint. In the capacitated lot-sizing model an additional constraint is introduced to limit the processing and setup time to the total capacity in the period (Quadt & Kuhn, 2008). To account for shipment coordination in an empirical application (i.e., the size of a transport fleet) constraint (26) is introduced to limit the total amount of production per period to maximum capacity Q_t.

$$\sum_i x_{i,t} \leq Q_t \quad \forall t \in T \tag{26}$$

In the static-dynamic model, constraint (18) limits the total amount of the order-up-to level in each period to a maximum capacity. This effectively limits the maximum possible production in a period. However, due to possible remaining inventory, the production quantity is often lower than the order up-to-level. Thus, the constraint is more strict here than in the static-capacity model where the production amount is directly limited. Consequently, the solution quality is affected by excluding solutions that require reordering on top of existing inventory.

3.4.2 Aggregate service level. Sereshti et al. (2021) introduce an aggregate service level constraint for the static model that is stricter than the individual service level constraints. They achieve a 1.5% lower cost, however, their computation time increases by four orders of magnitude. In the model, a quantile-based approach is used by defining a set of service levels K with an index k. The service levels become part of the decision variables. Constraint (4) is extended to constraint (28) by introducing a binary variable $l_{k,i}$ which indicates if the service level $\alpha_{k,i}$ is chosen for product i. Additionally, constraint (29) ensures that only one service level is chosen per item, and (30) is the domain constraint. Lastly, constraint (27) ensures that the weighted sum of all of the individual service levels does not exceed the service level α_a where $\sum_i w_i = 1$.

$$\sum_i \sum_k w_i \alpha_{k,i} l_{k,i} \geq \alpha_a \tag{27}$$

$$I_{i,0} + \sum_{j=1}^{t}(x_{i,j}) \geq F_{i,t}^{-1}(\alpha_{k,i}) l_{k,i} \qquad \forall k \in K, \forall i \in N, \forall t \in T \tag{28}$$

$$\sum_i l_{k,i} = 1 \qquad \forall k \in K \tag{29}$$

$$l_{k,i} \in \{0, 1\} \qquad \forall k \in K, \forall i \in N \tag{30}$$

3.5 Metaheuristic approach

To account for the increased solution time for larger instances, a metaheuristic approach is introduced for the static-dynamic model using a Variable Neighbourhood Descend (VND) approach. For each instance of the problem, we enforce the number of deliveries permitted n. For each item, a number of production cycles $X_{i,t,j}$ are determined with the production amount set larger than the expected demand for that cycle (31). To meet the service requirement, it is assumed that production always occurs in the first period. The cost of a cycle $c_{i,t,j}$ is determined as the sum of the setup cost f_i and holding costs h_i multiplied by the expected inventory in each cycle (32). However, in the metaheuristic, the cycle dependence is not directly enforced. In comparison, in the static-dynamic model, the constraint (18) ensures that the consecutive cycle order-up-to level is larger than the expected inventory level. Consequently, the cost function gives an approximation of the true objective value as the setup costs are incurred in each cycle.

$$S_{i,t,j} = G_{i,t,j-1}^{-1}(\alpha_i) \qquad \forall i \in N,\ t \in [1, T],\ j \in [t+1, T+1] \tag{31}$$

$$c_{i,t,j} = f_i + h_i \sum_{k=t}^{j-1}(S_{i,t,j} - \sum_{l=t}^{k} E[D_{i,l}]) \qquad \forall i \in N,\ t \in [1, T],\ j \in [t+1, T+1] \tag{32}$$

The problem is initialized with a solution of a single cycle starting at the first period and extending over the full planning horizon. After the initialization, the algorithm searches for improvements in the first neighborhood. If improvements are found, the steepest decent (highest reduced cost) is picked and the VND algorithm restarts from the first Neighbourhood. If no improvement is found, the search continues to the next neighborhood. After reaching the final neighborhood, the number of delivery periods n is extended by one. The consequent solution for the number of delivery periods n is recorded as the $BestSchedule_n$ and the search restarts from the first neighborhood. The algorithm continues until a solution is found for all numbers of shipments from one to T i.e., a shipment in every period.

Randa et al. (2019) show how search-based heuristics can be used in a static-dynamic lot-sizing model. Consequently, the following neighborhoods are used for the VND and the algorithm is described in Algorithm 1.

- Merge two cycles (Algorithm 3)
- Split a cycle in two without increasing the number of delivery periods (Algorithm 4)
- Shift the timing of a cycle for a single item (Algorithm 5)
- Shift the timing of all deliveries at a time (Algorithm 6)
- Split a cycle to increase the number of allowed shipments (Algorithm 2)

The full algorithm is described in Algorithm 1. The final neighborhood is described below (Algorithm 2) while the other neighborhoods are described in Appendix A.

Algorithm 1 The VND algorithm

Initialise:
terminate = False, *improvement* = False, n - number of shipments
$Schedule$ = None
for n *in* $range(T)$ **do**
    while not terminate **do**
        $Schedule, improvement \leftarrow MergeCycles(Schedule)$
        if not *improvement* **then**
            $Schedule, improvement \leftarrow ShiftTiming(Schedule)$
            if not *improvement* **then**
                $Schedule, improvement \leftarrow DivideCycle(Schedule)$
                if not *improvement* **then**
                    $Schedule, improvement \leftarrow ShiftTimingAllCycles(Schedule)$
                    if not *improvement* **then**
                        $BestSchedule_n = Schedule$
                        $Schedule \leftarrow IncreaseShipments(Schedule)$
                        terminate = True
                    end if
                end if
            end if
        end if
    end while
end for
return $BestSchedule_n \quad \forall n \in T$

Algorithm 2 Increase shipments

```
Input: Schedule
Initialise:
AllProductionPeriods = GetProductionPeriods(solution)
FreePeriods = range(T) not in AllProductionPeriods
for FreePeriod in FreePeriods do
    NewSchedule = Schedule
    for item, ProdPeriodsItem in Schedule do
        NewSchedule[item] = ProdPeriodsItem + FreePeriod
    end for
    if CalcCost(NewSchedule) < CalcCost(Schedule) then
        BestSchedule = NewSchedule
    end if
end for
return BestSchedule
```

4 Empirical example

4.1 Data

In papers considering numerical examples in stochastic lot-sizing, researchers often use a small number of periods (10 to 20) in the planning horizon and a similar amount of items (Sereshti et al., 2021; Tavaghof-Gigloo & Minner, 2021; Tempelmeier, 2007; Tempelmeier & Hilger, 2015). This thesis uses a small-scale data set with $N = 10$ items and $T = 12$ periods to simulate yearly planning. The period demand is generated similarly to Helber et al. (2013). The expected demands per item $E[D_i]$ are drawn from a discrete uniform distribution with a range $[150, 300]$. Further, the expected inter-period demand $E[D_{it}]$ for item i in period is drawn from a normal distribution with a mean value of $E[D_i]$ and standard deviation $E[D_i] \cdot V_{ip}$. Consequently, the inter-period demand is normally distributed with mean $E[D_{it}]$ a standard deviation $E[D_i] \cdot V_d$. Example item expected demands are shown in Table 3 and example period expected demands are shown in Table 4.

The holding costs h_i as set as consecutive integers $h_i = i \ \in \{1, 2, ..., N\}$. The setup costs f_i are dynamically set using the average expected demand for an item, the time between orders (TBO), and the holding costs (33) (Helber et al., 2013). Therefore the setup costs are not a direct model input but are kept at a ratio depending on the other model input parameters.

$$f_i = \frac{\frac{\sum_{t \in T} E[D_{i,t}]}{T} \cdot TBO^2 \cdot h_i}{2} \quad \forall i \in N \tag{33}$$

The emissions penalty P_t is determined as the sum of the setup costs in a period (34). This makes the emissions penalty dynamic relative to other parameters and gives a good middle ground between the emissions penalty not being impactful on the final solution and dominating the solutions.

$$P_t = \sum_{i \in N} f_i \quad \forall t \in T \tag{34}$$

The random elements are pseudo-randomly generated using the Python random library. The pseudo-random period demands are reused with each model, allowing the models to be fairly compared. In each further iteration, data sets are generated with sequential seeds, thus ensuring further reproducibility.

Table 3

Example expected demands for items $i \in N$

i	1	2	3	4	5	6	7	8	9	10
$E[D_i]$	165	200	173	292	246	191	254	245	294	252

Table 4

Example expected demands for items $i \in N$ in period $t \in T$

		t											
		1	2	3	4	5	6	7	8	9	10	11	12
i	1	194	317	171	182	154	131	192	171	168	179	153	211
	2	185	108	256	160	304	268	58	155	193	110	144	290
	3	263	174	204	149	97	201	162	250	128	175	231	187
	4	206	276	374	294	453	354	270	502	363	399	359	350
	5	192	348	191	304	298	345	203	291	149	248	313	212
	6	239	212	182	231	142	211	124	251	233	120	144	52
	7	286	173	95	268	273	258	152	330	263	327	108	279
	8	182	266	256	395	379	151	230	239	247	248	202	221
	9	309	192	433	389	271	464	385	300	50	99	149	369
	10	153	208	286	393	244	225	301	347	273	168	237	298

In the capacity-constrained models, the shipment capacity Q in a period is determined in equation (35) by the sum of all item expected demand ED_i times the capacity coefficient $q = 3$.

$$Q = \sum_{i \in N} E[D_i] \cdot q \tag{35}$$

For the aggregated service level, Sereshti et al. (2021) show that a set of 11 service level choices provides a good trade-off between computation time and accuracy. Therefore, the target aggregate service level of $\alpha_a = 0.95$ is chosen, and the set K of 11 service level options is equally distributed between 0.8 and 0.9999.

Lastly, in the ϵ-constraint method, the epsilon values are distributed between the objective value with the lowest cost and the largest number of shipments Z_1 and the objective value with the lowest emissions penalty and the least number of shipments Z_2. The number of chosen ϵ values determines the maximum number of different scenarios that are analyzed while increasing the solution time. Each added ϵ returns a solution with a number of shipments between the minimum and the maximum number of shipments. 12 evenly distributed ϵ values are chosen to show the optimum objective for each possible number of shipments. The μ value takes as $\mu = \frac{Z_2}{Z_1} \cdot 0.001$ to take account of the relative size between Z_2 and Z_1. M is taken as 100000 to enforce the set-up constraints. Table 5 shows the full overview of the used parameters.

Table 5

Parameters used

Description	
Number of items	$N = 10$
Number of periods	$T = 12$
Target service level for item i	$\alpha_i = 0.95$
Time between orders	$TBO = 3$
Inter-period variation of demand	$V_{ip} = 0.3$
Coefficient of variation of demand	$V_d = 0.3$
Holding cost for item i	$h_i = 1, 2, 3, ..., N$
Capacity coefficient	$q = 3$
Aggregate service level target	$\alpha_a = 0.95$
Product weight	$w_i = \frac{1}{N}$
Number of ϵ values chosen	12
Sufficiently large numbers	$M = 100000, M_2 = 20$
Sufficiently small number	$\mu = \frac{Z_2}{Z_1} \cdot 0.001$

5 Results

The obtained results are presented in the following section. Firstly, the performance of the models and the extensions are evaluated, followed by a cost-benefit analysis of the coordinated models against the uncoordinated ones. Secondly, the metaheuristic approach is assessed and compared to the static-dynamic method. Lastly, a sensitivity analysis is performed to demonstrate the consistency of the results.

5.1 Model evaluation

In this section, the model performance is evaluated. The models and their extensions were implemented in the commercial solver GUROBI version 10.0.1 using Python 3.11, solved with a 4 Core 2.30GHz processor using 8 threads and applied to the synthetic data set described in the previous section. Each model has 12 iterations; two to calculate the minimum and maximum boundary for the ϵ-constraint method, and a further 10 iterations for the ϵ values. Each iteration of the model, including every iteration of the augmented ϵ-constraint method, is given a 900-second time limit and an optimality gap of 0.001. The results are based on 100 simulations for all models except the static aggregate service levels model where only 10 simulations were performed due to the long run-time of the model.

All models, except the aggregated service level model, were solved until optimality in all iterations. Table 6 shows the number of shipments in the uncoordinated optimal solutions and the model statistics are presented in Table 7.

Table 6

The optimal number of shipments in the uncoordinated model

Uncoordinated optimal number of shipments	7	8	9	10	11	12	Total	Mean
Static	1	4	20	67	8		100	9.77
Static-dynamic		2	12	76	10		100	9.94
Static capacity		1	7	35	52	5	100	10.89
Static-dynamic capacity			6	43	43	8	100	10.53
Aggregate service levels	1	1	2	4	2		10	9.5
Receding horizon				2	19	79	100	11.77

Table 7

Model result overview

Model	Average simulation time (s)	Average time per iteration (s)	Solved to optimality	Average optimality gap	Worst optimality gap
Static	4.92	0.41	Yes	-	-
Static-dynamic	3.55	0.3	Yes	-	-
Receding-horizon (1st iteration)	5.1	0.43	Yes	-	-
Receding-horizon total	24.4	0.15	Yes	-	-
Static capacity	7.82	0.65	Yes	-	-
Static-dynamic capacity	13.4	1.12	Yes	-	-
Static aggregated service levels	8165.1	680.4	No	6.8%	14.6%

In all models, coordinating shipments allows a reduction in the number of shipments with a slight increase in the estimated cost while not breaching the service level requirements. In the static model, the uncoordinated solution required an average of 9.77 shipments. In each iteration, reducing the number of shipments to 4, a 59% average reduction in the number of shipments incurs an average 0.9% increase in cost. Similarly, in the static-dynamic model, where the uncoordinated optimal solution required an average of 9.94 shipments, a 59% reduction in the number of shipments to 4 shipments incurs a 2.0% increase in cost. This relation can be seen in Figure 1. Furthermore, Table 8 shows the average cost increase compared to the uncoordinated model solution. Since the static-dynamic model solutions have a lower cost than the static model, they are also more sensitive to cost increases. This also means that the static-dynamic model has a greater cost impact from shipment coordination. Additionally, as visible in Figure 1, the static-dynamic has an equivalent cost for a single shipment and provides an average of 11.5% reduction of cost for all other numbers of shipments. Thus the static-dynamic model outperformed the static model by having a lower computation time and lower cost.

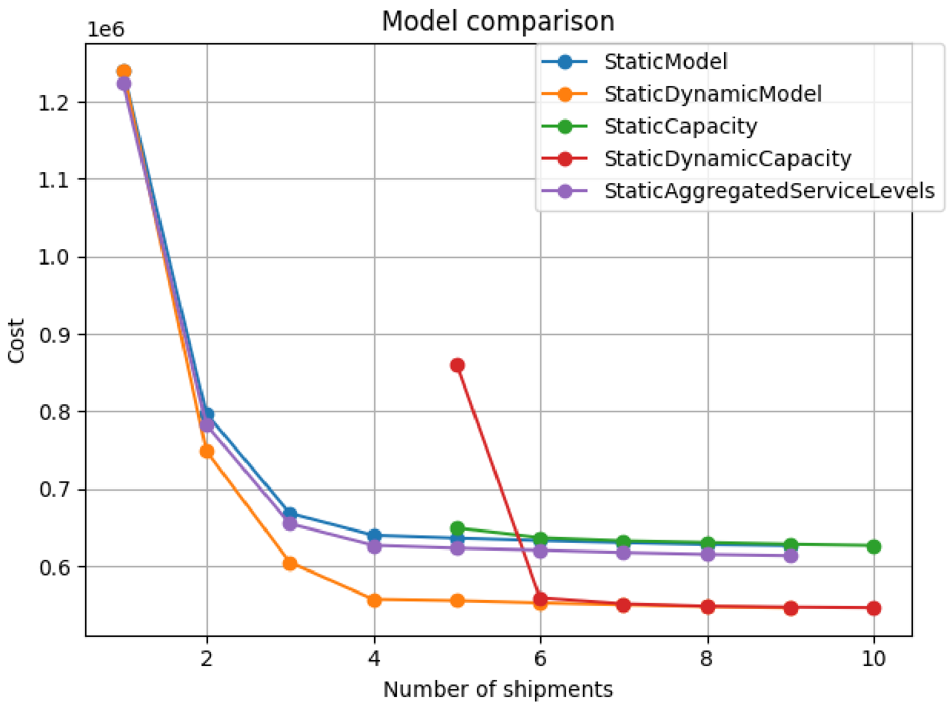


Figure 1. Model comparison

Table 8

Increase in cost compared to the uncoordinated solution

Number of shipments	Static	Static-dynamic	Static capacity	Static-dynamic capacity	Static aggregated service levels
11	0.0%	0.0%	0.0%	0.0%	0.0%
10	0.0%	0.0%	0.0%	0.0%	0.0%
9	0.0%	0.0%	0.0%	0.1%	0.0%
8	0.1%	0.3%	0.2%	0.2%	0.0%
7	0.2%	0.5%	0.3%	0.5%	0.0%
6	0.4%	1.0%	0.7%	1.1%	0.8%
5	0.6%	1.4%	1.4%	10.0%	1.3%
4	0.9%	2.0%			1.9%
3	3.2%	9.8%			6.39%
2	12.8%	35.9%			26.6%
1	51.0%	131.1%			104.4%

Similarly to the effect shown in Sereshti et al. (2021), the aggregated service levels reduced the cost at all numbers of shipments while maintaining the same service level. The reduction of the number of shipments is similar to the static model with a 55% reduction in the number of shipments incurring a 1.9% increase in cost. This is achieved even without achieving optimality at all numbers of shipments and with an increased solving time.

Further, when looking at the capacitated models, the uncoordinated solution of the static capacity model requires on average 1.12 more shipments than the static model. Similarly, the static-dynamic capacity model requires 0.57 more shipments than the uncapacitated model. In the coordinated approach, the capacitated static and static-dynamic models become infeasible with less than 5 shipments. However, the reduction in the number of shipments is still possible until that level. In the static model, a reduction in the number of shipments by 63.3% to 4 shipments can be achieved with a 1.4% increase in cost. Similarly, in the static-dynamic capacity model, a reduction in the average number of shipments of 52% to 5 shipments can be achieved with a 1.1% increase in cost. The models differ in the formulation of the capacity constraints i.e., in the static method the production amount is directly constrained while in the static-dynamic method the maximum order-up-to level is limited. The tighter bound of the dynamic model is highlighted in Figure 1 where at five shipments the dynamic solution provides a worse solution than the static method.

A simulation of the receding horizon approach can be seen in Figure 2. Here the period refers to the period where the production decisions are reevaluated. The graph shows the trade-off between cost and the number of shipments. The inclusion of the emissions penalty is visible in the convex shapes of the graphs. With the ϵ-constraint method, the model calculates solutions between the lowest cost and the lowest emission solution. In periods 10, 11, and 12 only a single solution is found as the lowest-cost solution is also the lowest emission solution. Since the receding horizon approach considers production in the current period, not the complete production plan, and incorporates realized demand, the model must produce in additional periods if necessary to meet the service level requirement. In this simulation, production occurred in periods 1, 3, 5, 7, and 9 compared to the expected 3 production periods in the initial production plan. However, this improves on the uncoordinated solution which has a mean of 11.77 shipments. Moreover, when looking at the solution values, in period 7, production occurred only for 2 out of the 10 items. Comparatively, in period 11, the model expected to produce in the following period, however, less demand was realized and no further production was needed. Nevertheless, since the receding horizon method determines the production on the lowest total cost (objective (24)), this effect can be mitigated by increasing the emission penalty term P. Moreover, the increase in the emission penalty term can be used to enforce a maximum required number of shipments while still allowing production deviations to meet the service level requirements. This can also be accomplished by using a nonlinear penalty that increases with the number of shipments.

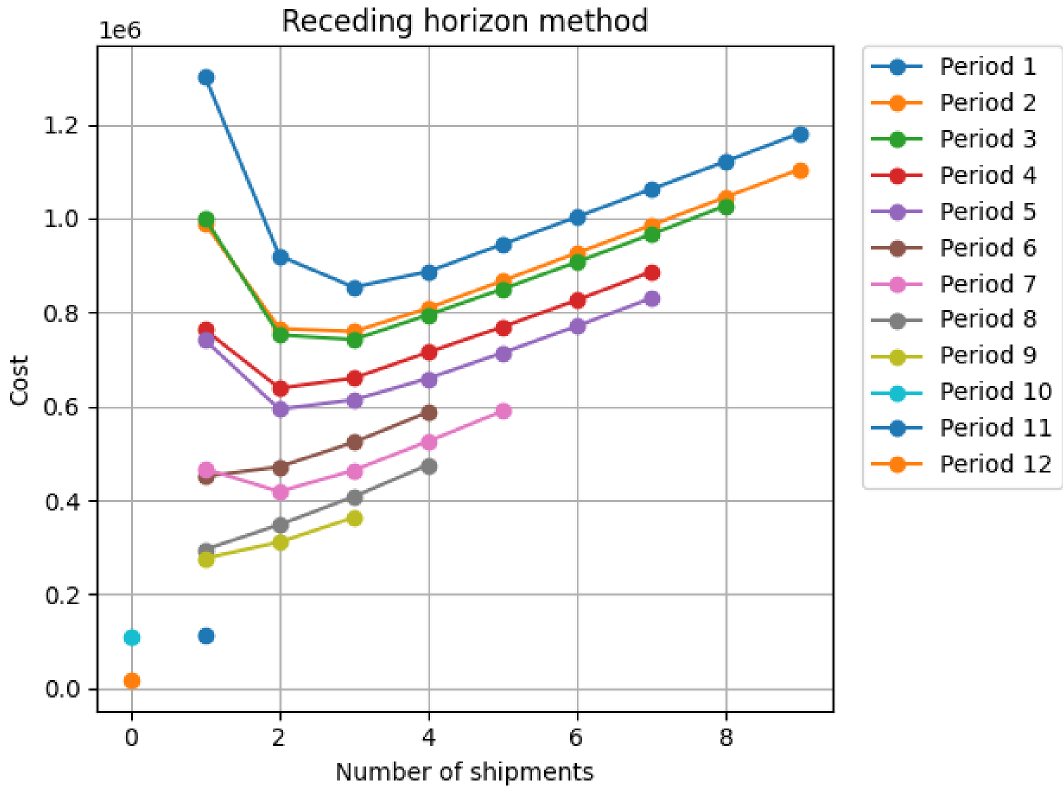


Figure 2. Receding horizon approach

5.2 Metaheuristic approach

In the following section, the metaheuristic approach is evaluated and compared with the results of the static-dynamic method. Over all of the iterations, the metaheuristic achieved less than 1.5% average optimality gap (Table 9). Smaller optimality gaps were achieved in the solutions with fewer shipments. Similarly to the static-dynamic model, the effects of coordinating shipments can be derived from the metaheuristic approach. In the heuristic, the average of the lowest cost number of shipments is 8.86. Further, in each iteration, the number of shipments can be reduced by an average of 54.8 % to 4 shipments, with a 1.94 % increase in cost. This effect can also be seen for a single simulation in Figure 3. Additionally, the run time was reduced by an order of magnitude (0.66 seconds for the heuristic versus 3.55 seconds for the static-dynamic model). Conclusively, the metaheuristic is a successful alternative to the optimal model by achieving comparative results while decreasing the total run-time.

Table 9

Metaheuristic average optimality gaps over 100 simulations

Number of shipments	1	2	3	4	5	6	7	8	9	10	11
Optimality gap (%)	0.00	0.00	0.44	0.77	0.56	0.71	0.87	1.01	1.20	1.32	1.27

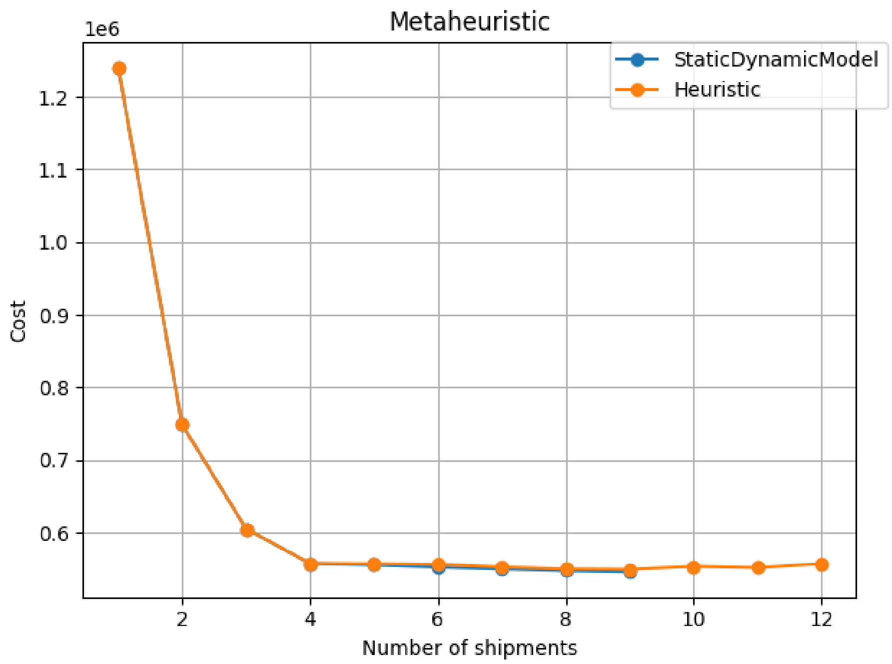


Figure 3. Metaheuristic comparison with the static-dynamic model

5.3 Sensitivity analysis

In the sensitivity analysis, all of the models are tested with differing parameters to show the consistency of the results. In each scenario, some model attributes are modified and the rest are kept constant to the initial model (Table 5). For each scenario, 10 synthetic data sets are generated and each data set is applied to all models. The time constraint per iteration is set to 60 seconds compared to 900 seconds in the full models. An exception was made for the scenario $N = 15, T = 18$ where the time limit was set to 120 seconds due to the aggregate service level model not finding an initial solution in 60 seconds. The complete list of scenarios is shown in Table 10 and the figures from all sensitivity analysis scenarios are shown in Appendix B.

Table 10

Sensitivity analysis scenario parameters

Scenario change	
Number of items	$N = \{5, 15\}$
Number of periods	$T = \{6, 18\}$
Number of items and periods	$N, T = \{(5, 6), (5, 18), (15, 6), (15, 18)\}$
Target service level for item i	$\alpha_i = \{0.90, 0.99\}$
Time between orders	$TBO = \{1, 5\}$
Inter-period variation of demand	$V_{ip} = \{0.1\}$
Coefficient of variation of demand	$V_d = \{0.1\}$
Holding cost for item i	$h_i = \{5.5, 5.5, ..., 5.5\}, \{1, 1, 1, 1, 1, 10, 10, 10, 10, 10\}$
Capacity coefficient	$q = \{2\}$
Aggregate service level target	$\alpha_a = \{0.90\}$

The model comparison with changes in the number of items and periods can be seen in Table 11. Increasing the number of items and the number of shipments significantly increases the run time. However, the static, static-dynamic, static-capacity, and receding horizon models achieved optimality in all scenarios. Furthermore, the aggregate service level model achieved optimality only in the smallest scenario while having a large optimality gap in the other scenarios.

The metaheuristic approach showed a significant run time decrease when compared with the static-dynamic model in all scenarios with an average optimality gap lower than 1.5%. This displays that the heuristic approach is useful when dealing with larger instances where the MIP models can not achieve optimality in a feasible time. The effects of additional items and periods with shipment coordination can be seen in Figure 4. Because of the larger planning horizon, in the static and static-dynamic models, a

62.5% reduction in the number of shipments can be made with a 2% increase in cost. However, with a capacity limitation, the coordination opportunities are limited as it requires a few large shipments that are constrained by the available capacity.

Moreover, Figure 5 and Figure 6 show the effect of changing time between orders and by proxy changing the setup costs. In Figure 5 where setup costs are lower, the marginal cost of decreasing the number of shipments is increased. Comparatively, in Figure 6 where the setup costs are high, the marginal increase of total costs when decreasing the number of shipments becomes minimal. This shows that the impact of coordination has a lower downside in situations with high setup costs. However, when adjusting the holding costs per item there is no significant change to the cost of shipment coordination as seen in Figure 7 and Figure 8.

Conclusively, the sensitivity analysis shows that the models are consistent with changes in the number of periods, the number of items, and variations in the input parameters.

Table 11

Sensitivity analysis model comparison

Scenario		Static	Static-dynamic	Static capacity	Static-dynamic capacity	Aggregated service levels	Receding horizon	Metaheuristic
$N = 5$	Runtime (s)	2.85	1.80	4.01	4.98	480.45	9.72	0.30
	Avg opt gap (%)	-	-	-	-	4.83	-	0.97
$N = 15$	Runtime (s)	6.91	3.98	12.20	27.65	669.48	17.39	0.35
	Avg opt gap (%)	-	-	-	-	14.25	-	0.79
$T = 6$	Runtime (s)	0.63	0.37	1.59	1.61	455.60	1.41	0.11
	Avg opt gap (%)	-	-	-	-	2.80	-	0.46
$T = 18$	Runtime (s)	14.18	12.71	17.57	52.43	677.11	58.43	1.06
	Avg opt gap (%)	-	-	-	0.36	8.43	-	0.95
$N = 5, T = 6$	Runtime (s)	0.24	0.42	0.45	0.60	36.96	0.87	0.06
	Avg opt gap (%)	-	-	-	-	-	-	0.36
$N = 15, T = 6$	Runtime (s)	0.66	1.67	1.98	2.01	536.26	1.71	0.18
	Avg opt gap (%)	-	-	-	-	4.00	-	0.48
$N = 15, T = 18$	Runtime (s)	62.31	45.93	82.59	172.99	1364.33	295.60	5.89
	Avg opt gap (%)	-	-	-	1.44	6.55	-	0.62
$N = 5, T = 18$	Runtime (s)	14.88	16.42	19.85	79.02	523.26	62.97	1.23
	Avg opt gap (%)	-	-	-	0.06	5.31	-	1.08

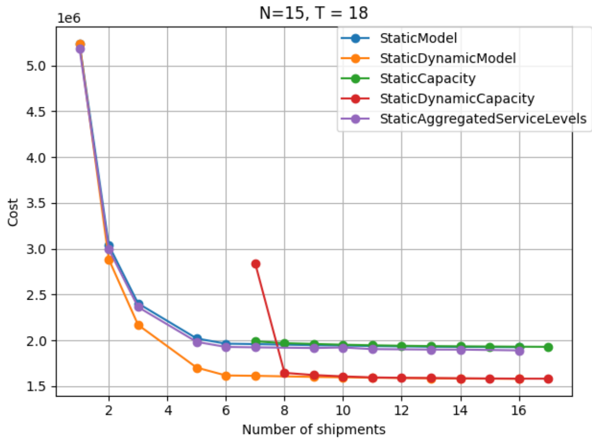


Figure 4. Scenario $N = 15$ $T = 18$

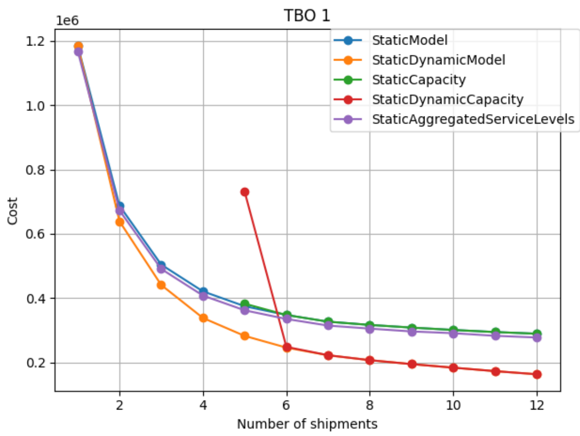


Figure 5. Scenario $TBO = 1$

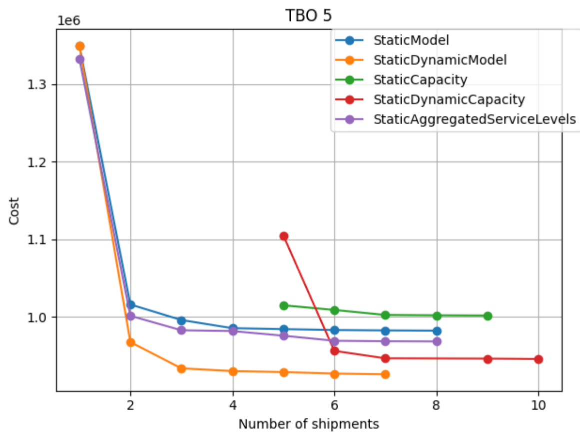


Figure 6. Scenario $TBO = 5$

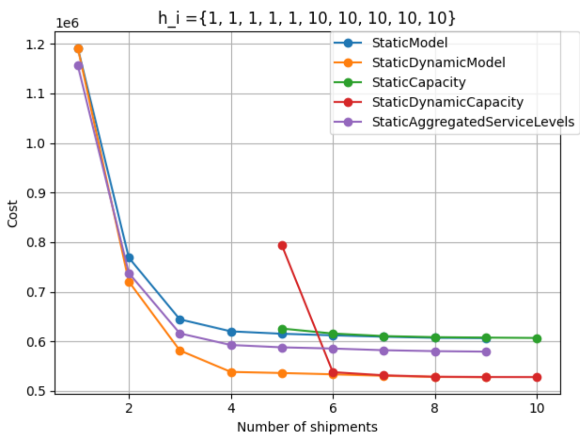


Figure 7. Scenario

$h_i = \{1, 1, 1, 1, 1, 10, 10, 10, 10, 10\}$

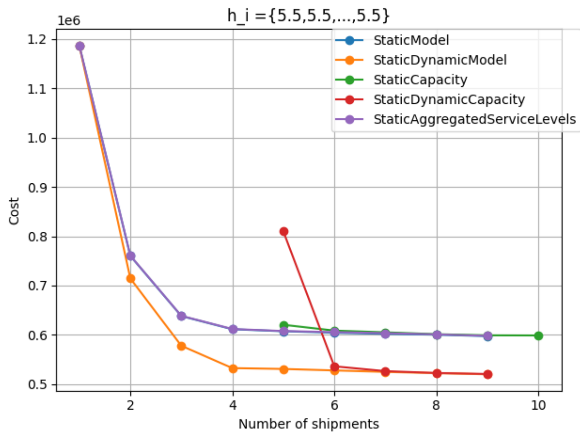


Figure 8. Scenario $h_i = \{5.5, 5.5, ..., 5.5\}$

6 Conclusion

This thesis investigates reducing emissions by coordinating shipments in the multi-item stochastic lot-sizing problem. Moreover, this thesis analyses the trade-offs between cost minimization and emission reduction. The models use the augmented ϵ-constraint bi-objective modeling and analyze static, static-dynamic, and static receding horizon methods of demand uncertainty. Additionally, extensions with capacity limitations and aggregated service levels are considered. The theis shows that a significant reduction of periods with shipments can be achieved with a slight increase in total cost. This provides a cost-efficient way to reduce carbon footprint in transportation.

Next, a variable neighborhood descent meta-heuristic was introduced for the static-dynamic method. The metaheuristic achieved an average optimality gap of 0.74% while reducing the run time by 82.3% compared to the static-dynamic method. This shows that the metaheuristic is a useful alternative to the mixed integer programming formulation of the static-dynamic method.

Finally, a sensitivity analysis was performed for all models showing that the models are consistent when the parameters and scale of the models are varied. Furthermore, the sensitivity analysis showed that the coordination of shipments has lower marginal costs in scenarios with high setup costs.

Consequently, this thesis shows how operations research modeling can be used to progress towards sustainability goals. The shipment coordination and similar methods can be effectively used in supply chain management to reduce emissions with a minor cost increase. These methods can also be further integrated into decision-support systems.

Since the analysis in this thesis was performed using a random, synthetically generated data set, in further research, the models introduced in this thesis can be tested using real empirical data. Additionally, the metaheuristic approach considers the uncapacitated static-dynamic model and can be expanded using capacity limitations, minimum order requirements, or other extensions. Similarly, shipment coordination can be applied using dynamic uncertainty where the production decisions and the production quantities are recalculated every period.

References

Aloulou, M. A., Dolgui, A., & Kovalyov, M. Y. (2014). A bibliography of non-deterministic lot-sizing models. *International Journal of Production Research*, *52*(8), 2293–2310.

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

Carrillo, M. (2022). Measuring progress towards sustainability in the european union within the 2030 agenda framework. *Mathematics*, *10*(12), 2095.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, *8*(3), 631–657.

Diaz, D., & Moore, F. (2017). Quantifying the economic risks of climate change. *Nature Climate Change*, *7*(11), 774–782.

Dural-Selcuk, G., Rossi, R., Kilic, O. A., & Tarim, S. A. (2020). The benefit of receding horizon control: Near-optimal policies for stochastic inventory control. *Omega*, *97*, 102091.

Forel, A., & Grunow, M. (2022). Dynamic stochastic lot sizing with forecast evolution in rolling-horizon planning. *Production and Operations Management*.

Gao, L.-L., Altay, N., & Robinson, E. P. (2008). A comparative study of modeling and solution approaches for the coordinated lot-size problem with dynamic demand. *Mathematical and Computer Modelling*, *47*(11-12), 1254–1263.

Glock, C. H., Grosse, E. H., & Ries, J. M. (2014). The lot sizing problem: A tertiary study. *International Journal of Production Economics*, *155*, 39–51.

Harris, F. W. (1913). How many parts to make at once. *The Magazine of Management*, *10*(2).

Heck, M., & Schmidt, G. (2010). Lot-size planning with non-linear cost functions supporting environmental sustainability. *International Journal of Green Computing (IJGC)*, *1*(1), 34–39.

Helber, S., Sahling, F., & Schimmelpfeng, K. (2013). Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR spectrum*, *35*, 75–105.

Hua, G., Cheng, T., & Wang, S. (2011). Managing carbon footprints in inventory management. *International journal of production economics*, *132*(2), 178–185.

Kadziński, M., Tervonen, T., Tomczyk, M. K., & Dekker, R. (2017). Evaluation of multi-objective optimization approaches for solving green supply chain design problems. *Omega*, *68*, 168–184.

Khaled, M. S., Shaban, I. A., Karam, A., Hussain, M., Zahran, I., & Hussein, M. (2022). An analysis of research trends in the sustainability of production planning. *Energies*, *15*(2), 483.

Kim, I. Y., & De Weck, O. L. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, *29*, 149–158.

Liu, C.-H. (2016). Discrete lot-sizing and scheduling problems considering renewable energy and co 2 emissions. *Production Engineering*, *10*, 607–614.

Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, *213*(2), 455–465.

Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ε-constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, *219*(18), 9652–9669.

McKinnon, A., Browne, M., Whiteing, A., & Piecyk, M. (2015). *Green logistics: Improving the environmental sustainability of logistics*. Kogan Page Publishers.

Nikas, A., Fountoulakis, A., Forouli, A., & Doukas, H. (2022). A robust augmented ε-constraint method (augmecon-r) for finding exact solutions of multi-objective linear programming problems. *Operational research*, 1–42.

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

Quadt, D., & Kuhn, H. (2008). Capacitated lot-sizing with extensions: A review. *4OR*, *6*(1), 61–83.

Randa, A. C., Doğru, M. K., Iyigun, C., & Özen, U. (2019). Heuristic methods for the capacitated stochastic lot-sizing problem under the static-dynamic uncertainty strategy. *Computers & Operations Research*, *109*, 89–101.

Retel Helmrich, M. J., Jans, R., van den Heuvel, W., & Wagelmans, A. P. (2015). The economic lot-sizing problem with an emission capacity constraint. *European Journal of Operational Research*, *241*(1), 50–62.

Rizet, C., Cruz, C., & de Lapparent, M. (2014). Co2 emissions of french shippers: The roles of delivery frequency and weight, mode choice, and distance. *Research in Transportation Business & Management*, *12*, 20–28.

Robinson, P., Narayanan, A., & Sahin, F. (2009). Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, *37*(1), 3–15.

Romeijn, H. E., Morales, D. R., & Van den Heuvel, W. (2014). Computational complexity of finding pareto efficient outcomes for biobjective lot-sizing models. *Naval Research Logistics (NRL)*, *61*(5), 386–402.

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

Sadjadi, S. J., Heidari, M., & Alinezhad Esboei, A. (2014). Augmented ε-constraint method in multiobjective staff scheduling problem: A case study. *The International Journal of Advanced Manufacturing Technology*, *70*, 1505–1514.

Sepehri, A., & Gholamian, M. R. (2022). Joint pricing and lot-sizing for a production model under controllable deterioration and carbon emissions. *International Journal of Systems Science: Operations & Logistics*, *9*(3), 324–338.

Sereshti, N., Adulyasak, Y., & Jans, R. (2021). The value of aggregate service levels in stochastic lot sizing problems. *Omega*, *102*, 102335.

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

Tavaghof-Gigloo, D., & Minner, S. (2021). Planning approaches for stochastic capacitated lot-sizing with service level constraints. *International Journal of Production Research*, *59*(17), 5087–5107.

Tempelmeier, H. (2007). On the stochastic uncapacitated dynamic single-item lotsizing problem with service level constraints. *European Journal of Operational Research*, *181*(1), 184–194.

Tempelmeier, H. (2011). A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. *Omega*, *39*(6), 627–633.

Tempelmeier, H., & Herpers, S. (2010). Abc β–a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. *International Journal of Production Research*, *48*(17), 5181–5193.

Tempelmeier, H., & Herpers, S. (2011). Dynamic uncapacitated lot sizing with random demand under a fillrate constraint. *European Journal of Operational Research*, *212*(3), 497–507.

Tempelmeier, H., & Hilger, T. (2015). Linear programming models for a stochastic dynamic capacitated lot sizing problem. *Computers & Operations Research*, *59*, 119–125.

Tunc, H., Kilic, O. A., Tarim, S. A., & 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.

United Nations. (2015). Transforming our world: The 2030 agenda for sustainable development. https://sdgs.un.org/2030agenda

Vaez, P., Sabouhi, F., & Jabalameli, M. S. (2019). Sustainability in a lot-sizing and scheduling problem with delivery time window and sequence-dependent setup cost consideration. *Sustainable Cities and Society*, *51*, 101718.

van den Heuvel, W., Romeijn, H. E., & Morales, D. R. (2012). Bi-objective economic lot-sizing problems. *IWLS 2012*, 82.

Wang, M., Wood, L. C., & Wang, B. (2022). Transportation capacity shortage influence on logistics performance: Evidence from the driver shortage. *Heliyon*, *8*(5), e09423.

Yağmur, E., & Kesen, S. E. (2023). Bi-objective coordinated production and transportation scheduling problem with sustainability: Formulation and solution approaches. *International Journal of Production Research*, *61*(3), 774–795.

Zhang, H., Razzaq, A., Pelit, I., & Irmak, E. (2022). Does freight and passenger transportation industries are sustainable in brics countries? evidence from advance panel estimations. *Economic research-Ekonomska istraživanja*, *35*(1), 3690–3710.

Appendix A

VND neighbourhoods

Algorithm 3 Merge cycles

Input: $Schedule$
Initialise:
$improvement = False$
for $item, ProdPeriodsItem$ in $Schedule$ **do**
    $NewSchedule = Schedule$
    for i in $range(len(ProdPeriodsItem) - 1)$ **do**
        $(a, b) = ProdPeriodsItem[i]$
        $(c, d) = ProdPeriodsItem[i + 1]$
        $NewSchedule[item] = ProdPeriodsItem[: i] + (a, d) + ProdPeriodsItem[i + 1 :]$
        if $GetNumPeriods(NewSchedule) == GetNumPeriods(Schedule)$ **then**
            if $CalcCost(NewSchedule) < CalcCost(Schedule)$ **then**
                $improvement = True$
                $Schedule = NewSchedule$
            end if
        end if
    end for
end for
return $Schedule, improvement$

Algorithm 4 Split cycles

```
Input: Schedule
Initialise:
improvement = False
AllProductionPeriods = GetProductionPeriods(solution)
for item, ProdPeriodsItem in Schedule do
    NewSchedule = Schedule
    FreePeriods = AllProductionPeriods not in ProdPeriodsItem
    Combinations = GetPossibleCombinations(FreePeriods)
    for NewPeriods in Combinations do
        NewSchedule[item] = ProdPeriodsItem + NewPeriods
        if CalcCost(NewSchedule) < CalcCost(Schedule) then
            improvement = True
            Schedule = NewSchedule
        end if
    end for
end for
return Schedule, improvement
```

Algorithm 5 Shift cycles

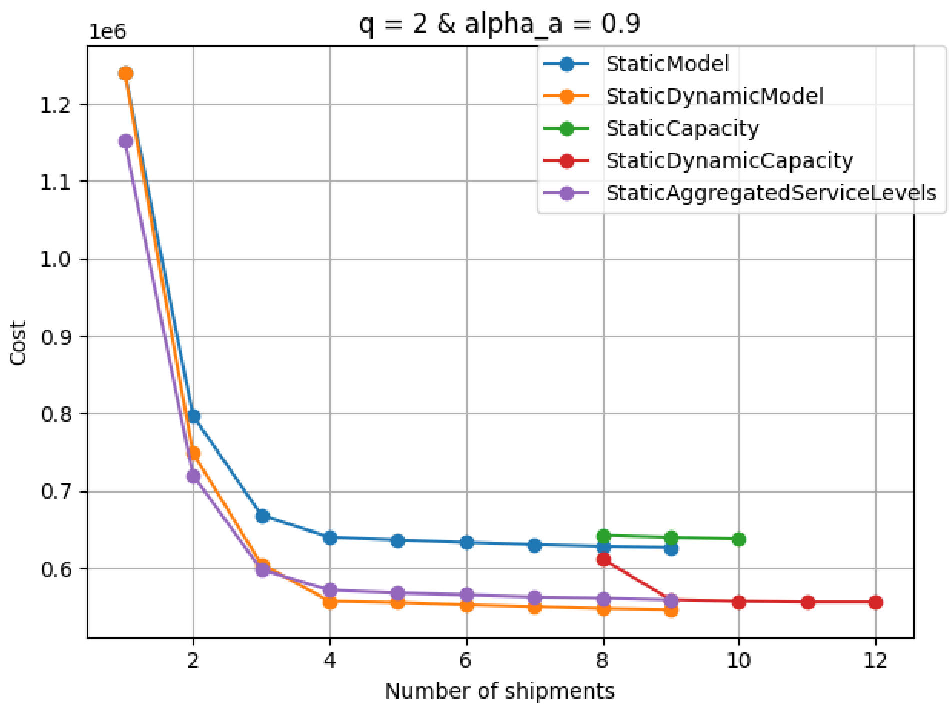
```
Input: Schedule
Initialise:
improvement = False
AllProductionPeriods = GetProductionPeriods(solution)
for item, ProdPeriodsItem in Schedule do
    NewSchedule = Schedule
    FreePeriods = AllProductionPeriods not in ProdPeriodsItem
    ShiftPeriods = ProdPeriodsItem − [FirstPeriod, LastPeriod]
    for FreePeriod in FreePeriods do
        for ShiftPeriod in ShiftPeriods do
            NewSchedule[item] = ProdPeriodsItem − ShiftPeriod + FreePeriod
            if CalcCost(NewSchedule) < CalcCost(Schedule) then
                improvement = True
                Schedule = NewSchedule
            end if
        end for
    end for
end for
return Schedule, improvement
```

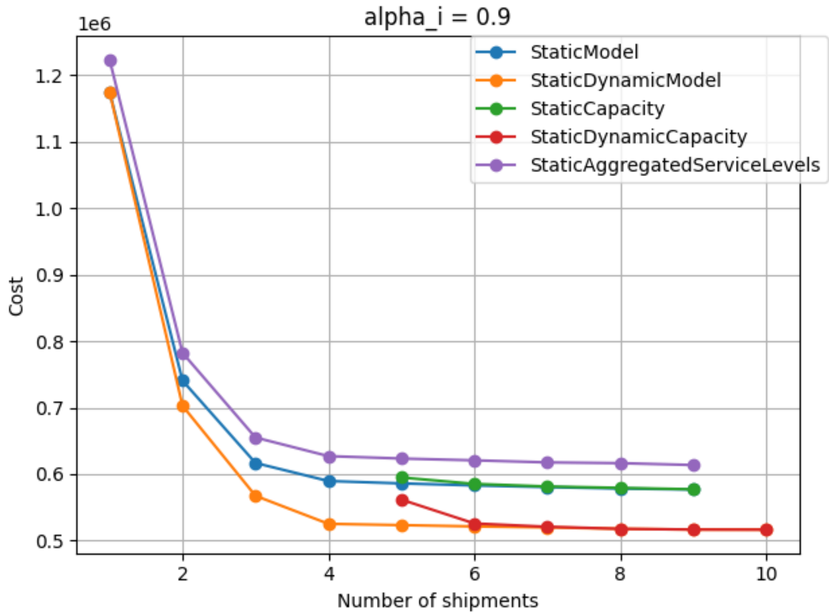
Algorithm 6 Shift all cycles

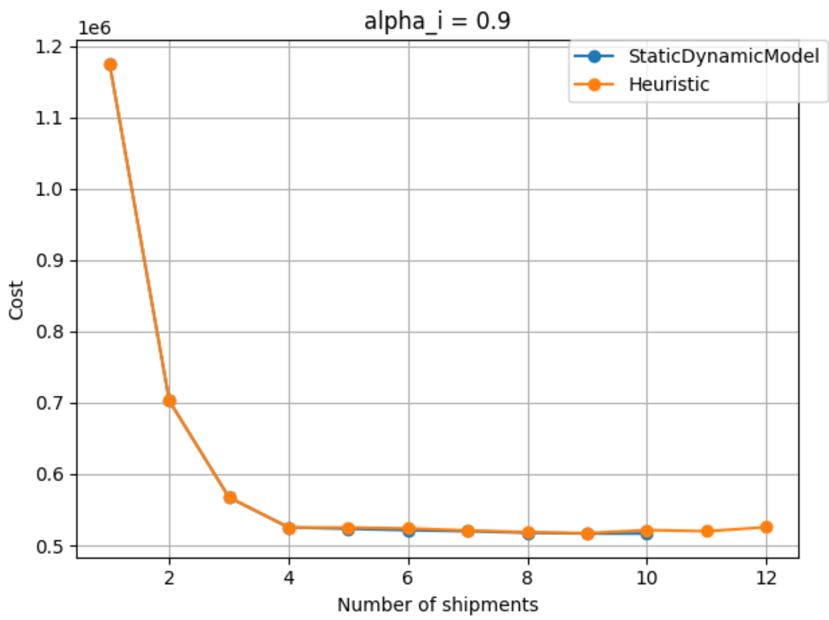
```
Input: Schedule
Initialise:
improvement = False
AllProductionPeriods = GetProductionPeriods(solution)
FreePeriods = range(T) not in ProdPeriodsItem
ShiftPeriods = ProdPeriodsItem − [FirstPeriod, LastPeriod]
for FreePeriod in FreePeriods do
    for ShiftPeriod in ShiftPeriods do
        NewSchedule = Schedule
        for item, ProdPeriodsItem in Schedule do
            if ShiftPeriod in ProdPeriodsItem then
                NewSchedule[item] = ProdPeriodsItem − ShiftPeriod + FreePeriod
            end if
        end for
        if CalcCost(NewSchedule) < CalcCost(Schedule) then
            improvement = True
            Schedule = NewSchedule
        end if
    end for
end for
return Schedule, improvement
```

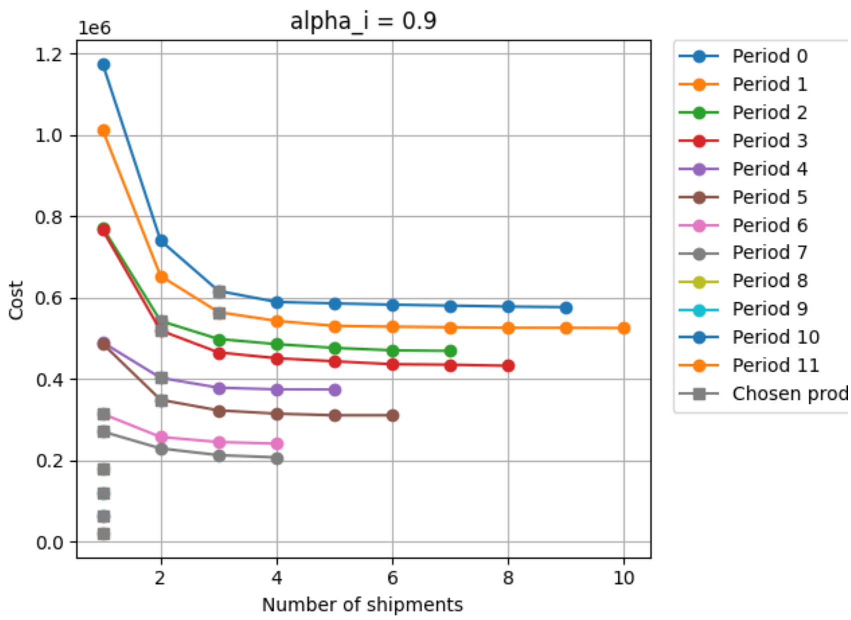
Appendix B

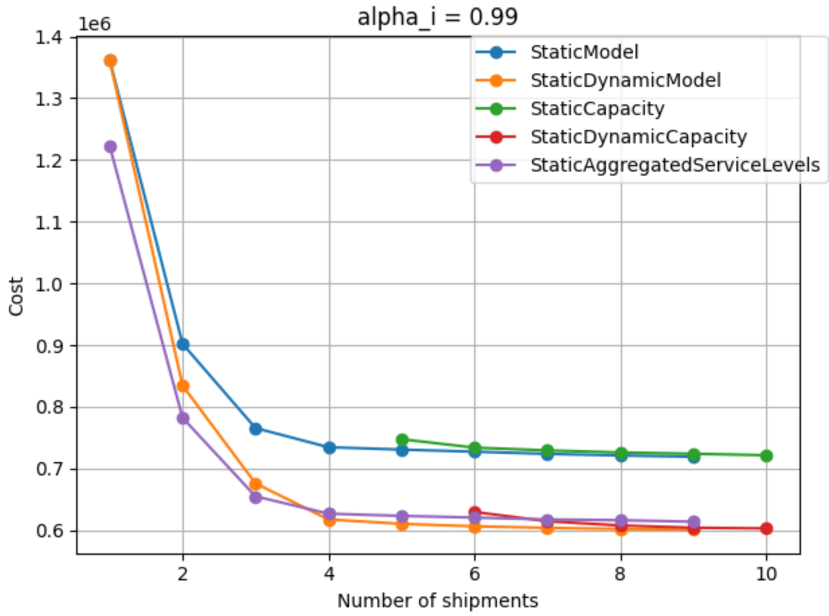
Sensitivity analysis figures

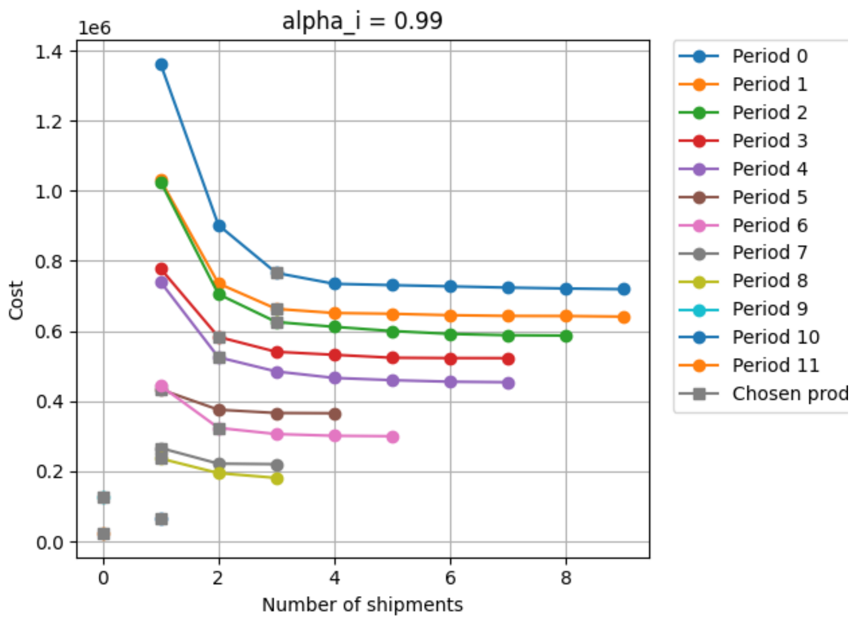


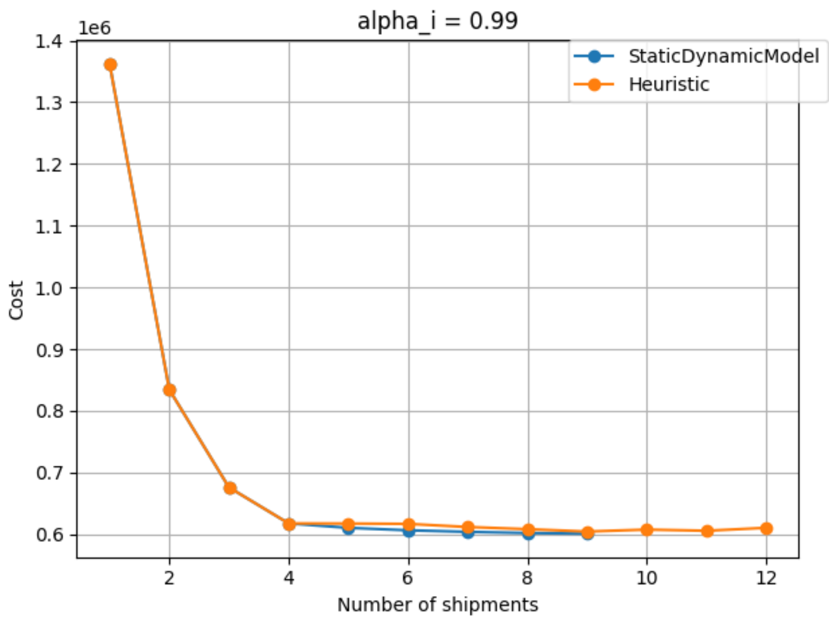

alpha_i = 0.9
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels

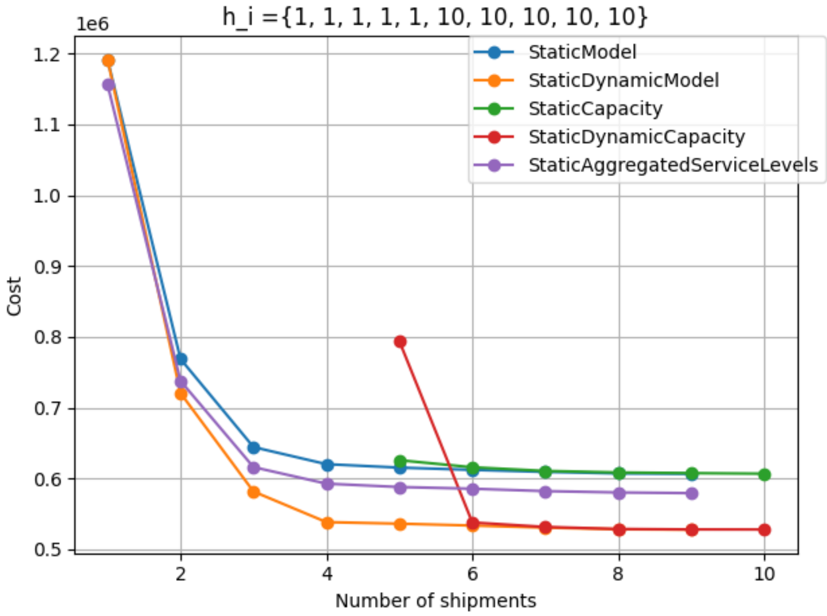

alpha_i = 0.9
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic

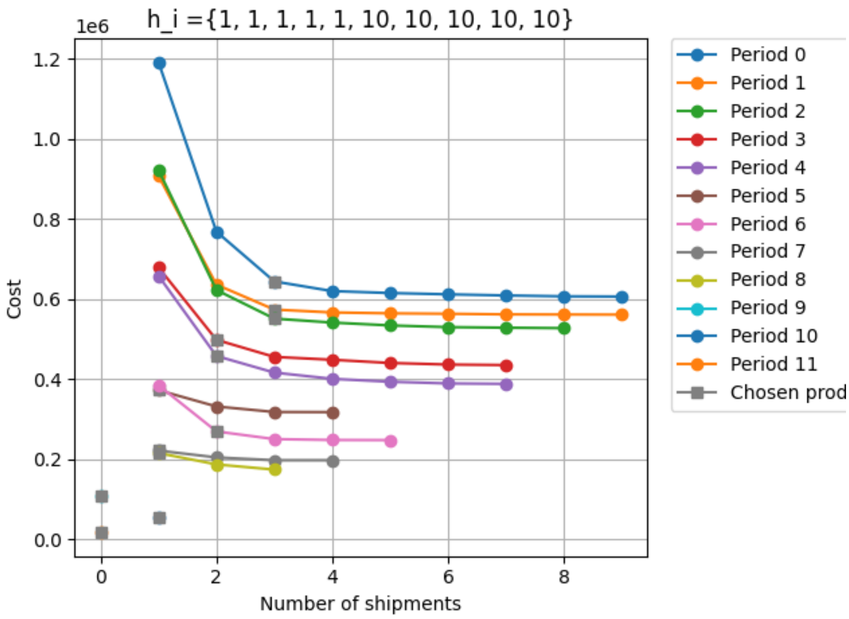

alpha_i = 0.9
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod

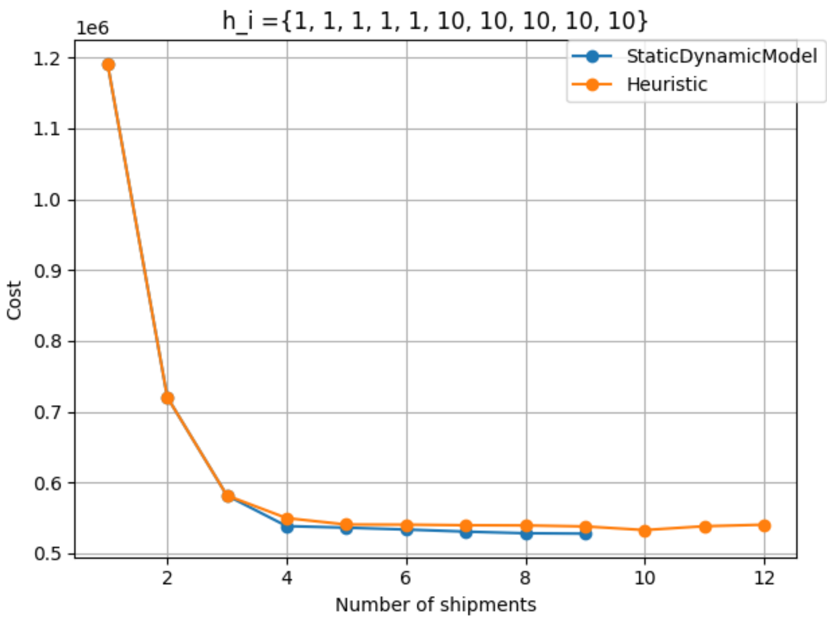

alpha_i = 0.99
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels

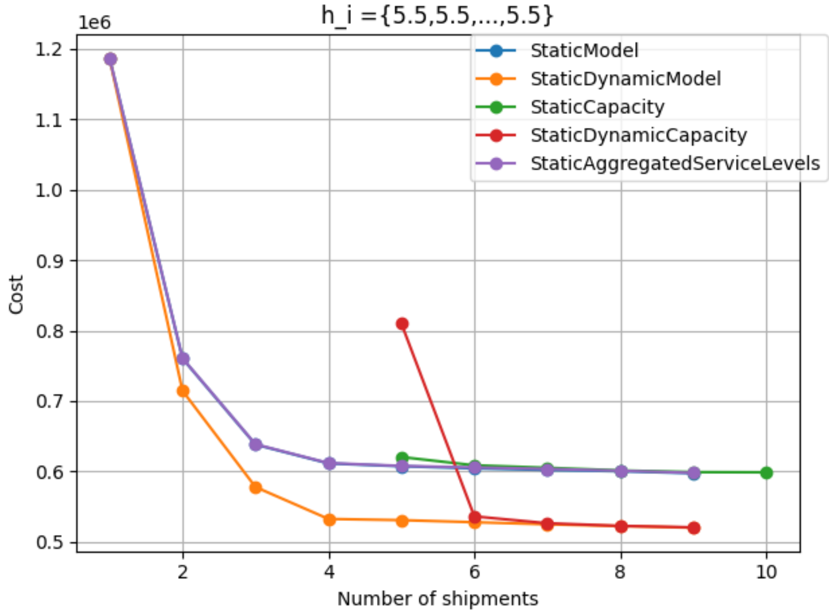

alpha_i = 0.99
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod

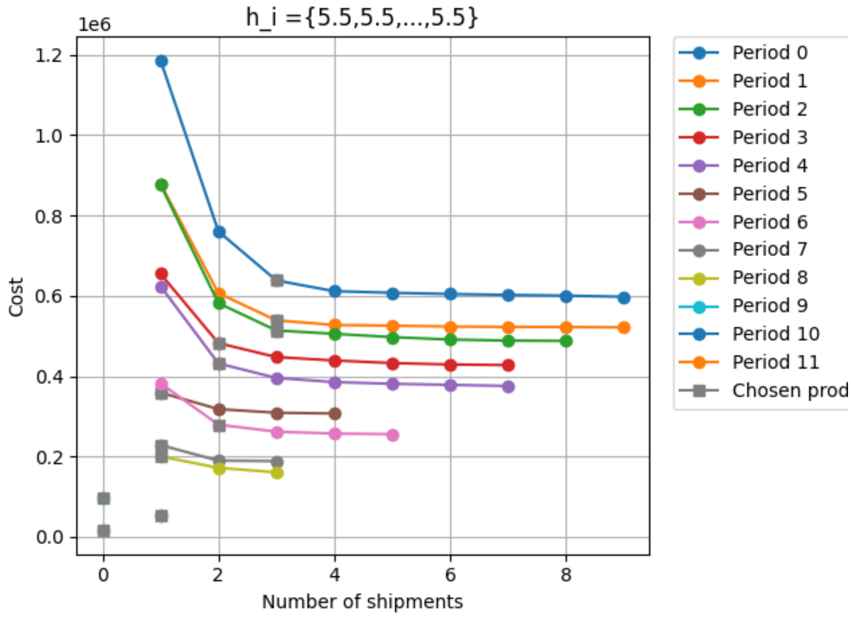

alpha_i = 0.99
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic

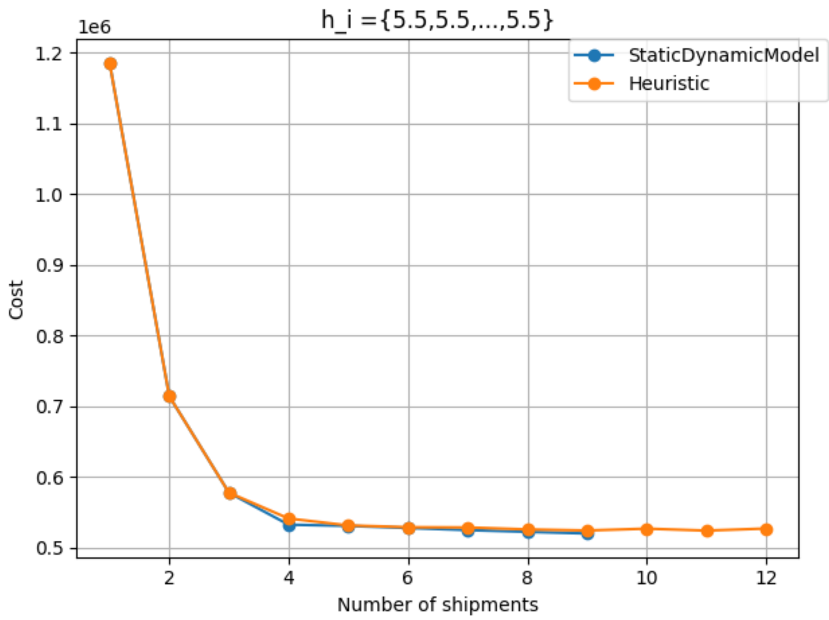

h_i ={1, 1, 1, 1, 1, 10, 10, 10, 10, 10}
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels

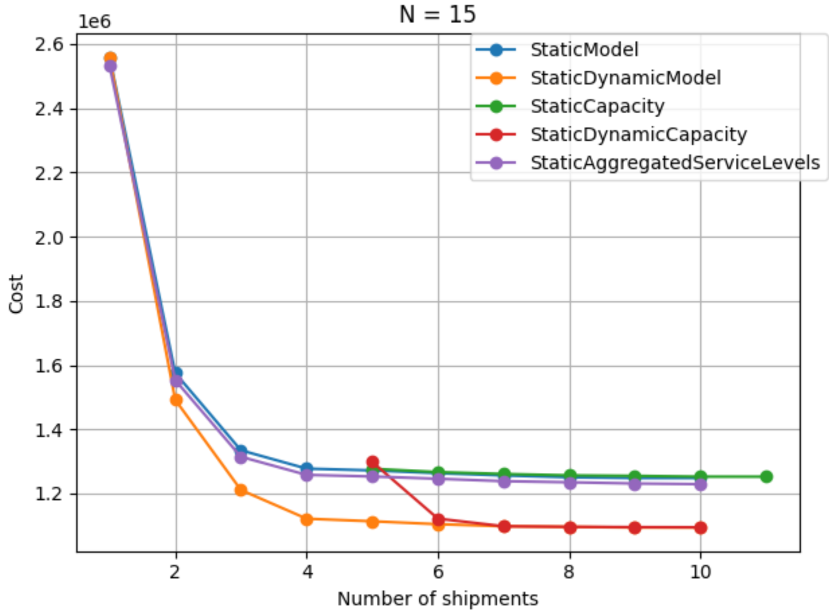

h_i ={1, 1, 1, 1, 1, 10, 10, 10, 10, 10}
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod

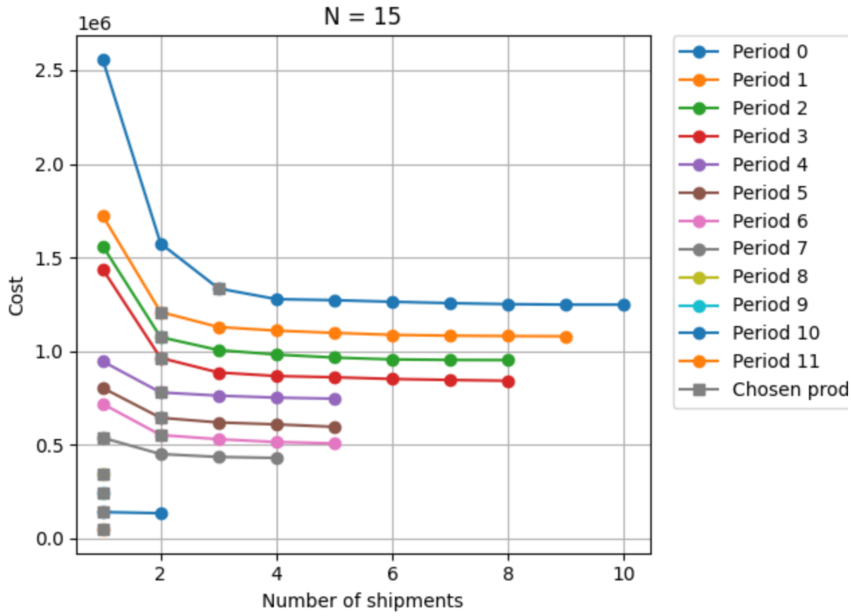

h_i ={1, 1, 1, 1, 1, 10, 10, 10, 10, 10}
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic

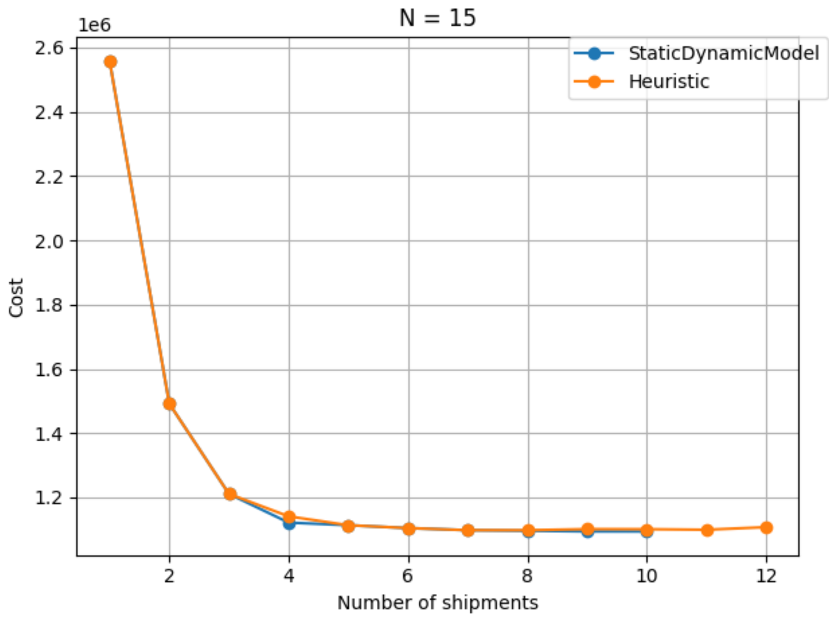

h_i ={5.5,5.5,...,5.5}
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels

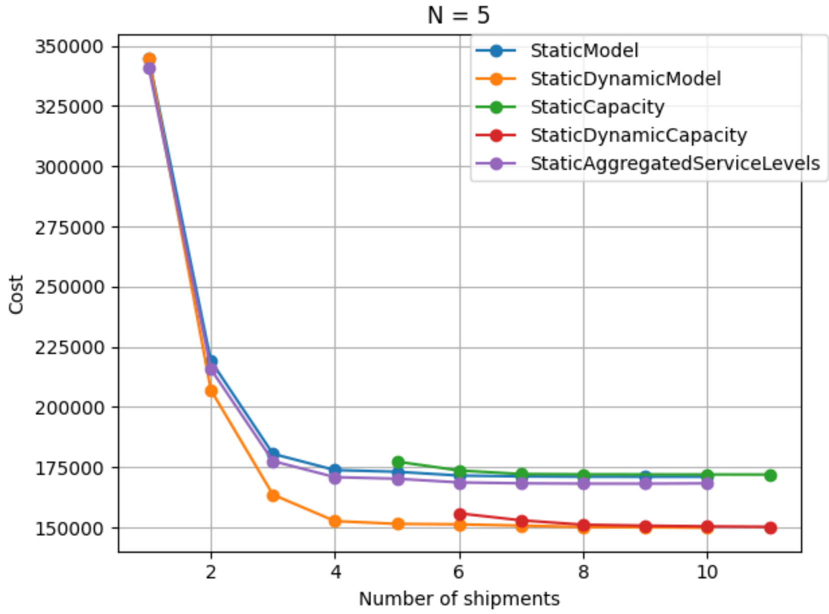

h_i ={5.5,5.5,...,5.5}
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod

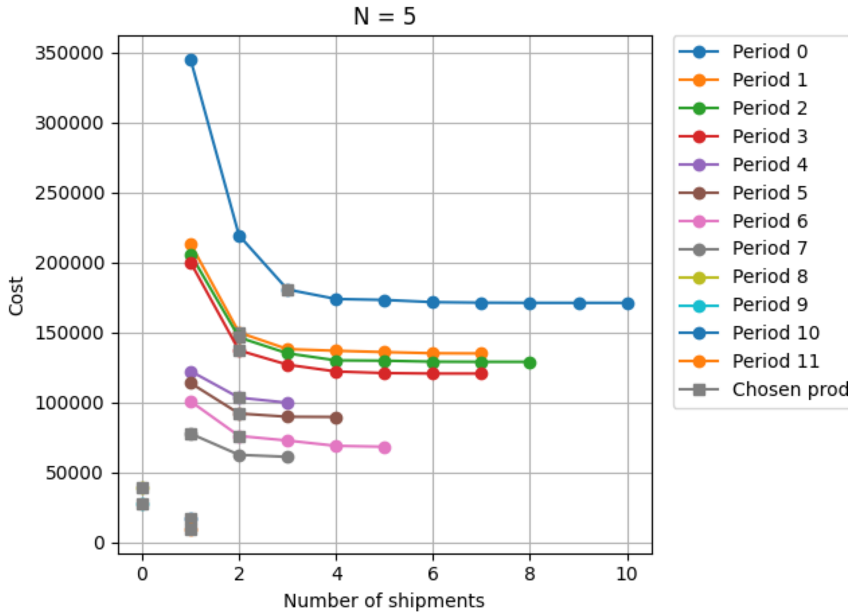

h_i ={5.5,5.5,...,5.5}
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic

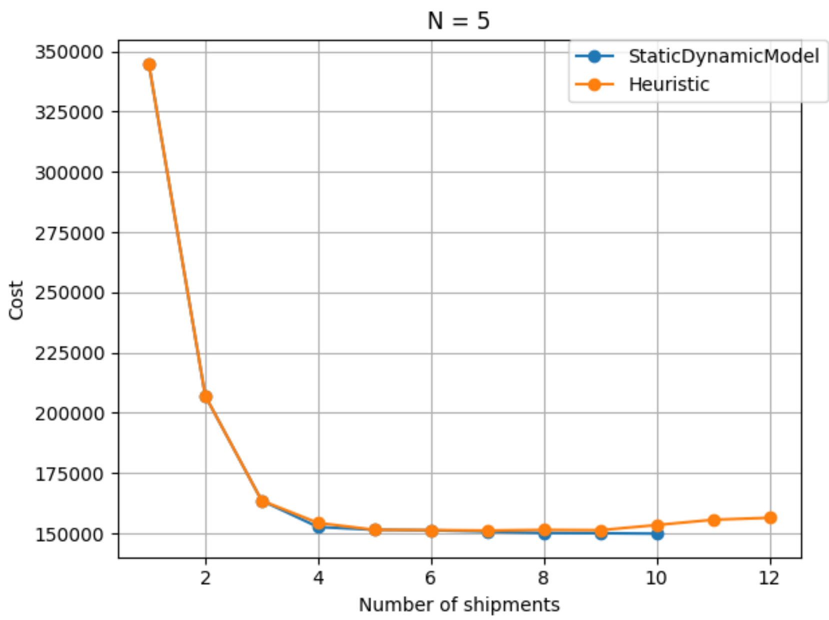

N = 15
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels

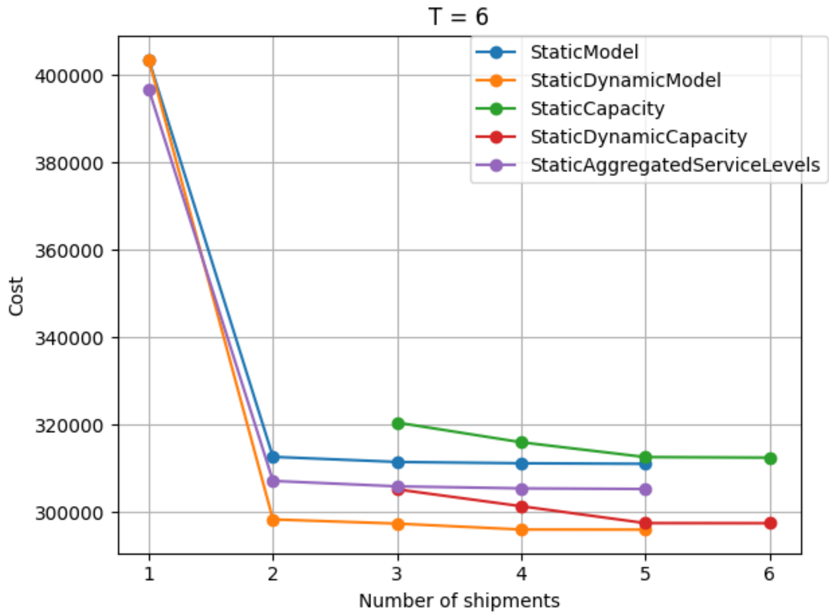

N = 15
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod

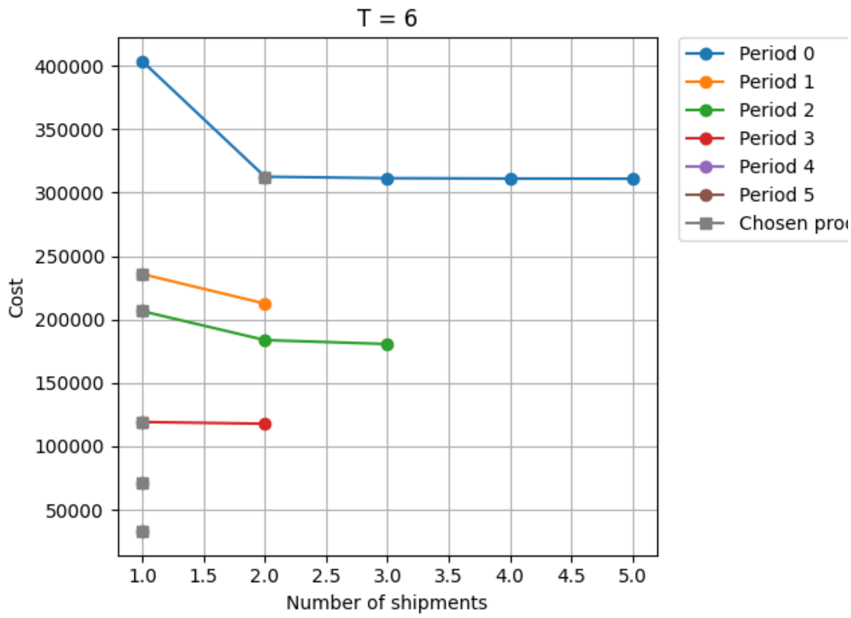

N = 15
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic

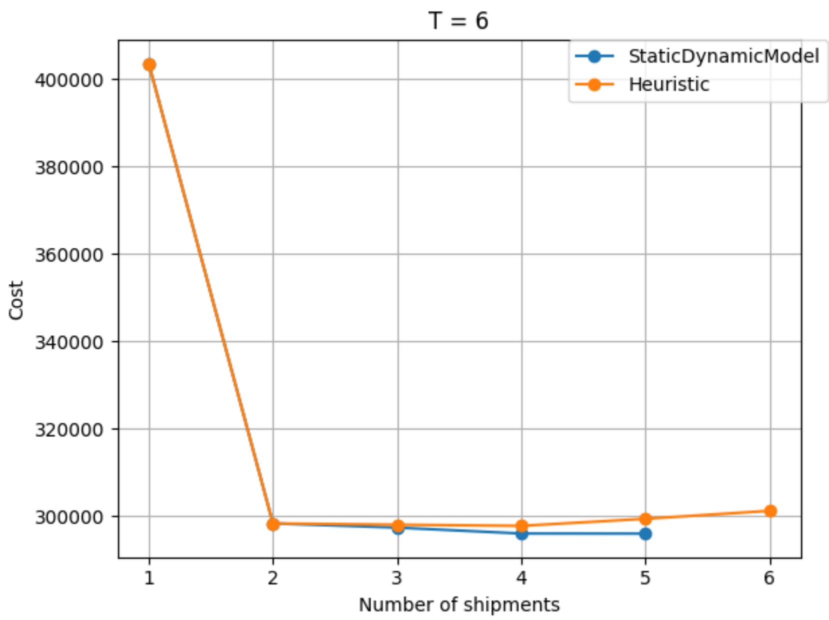

N = 5
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments

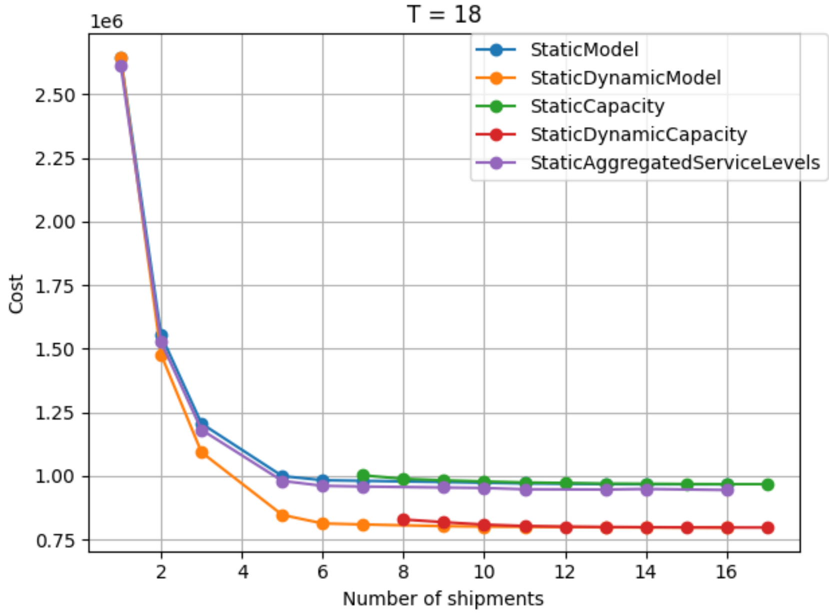

N = 5
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod
Cost
Number of shipments

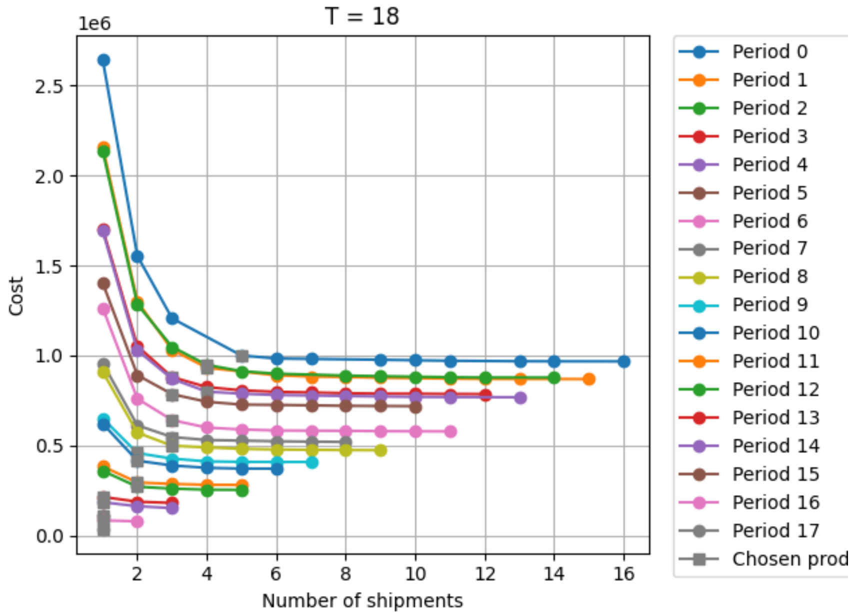

N = 5
StaticDynamicModel
Heuristic
Cost
Number of shipments

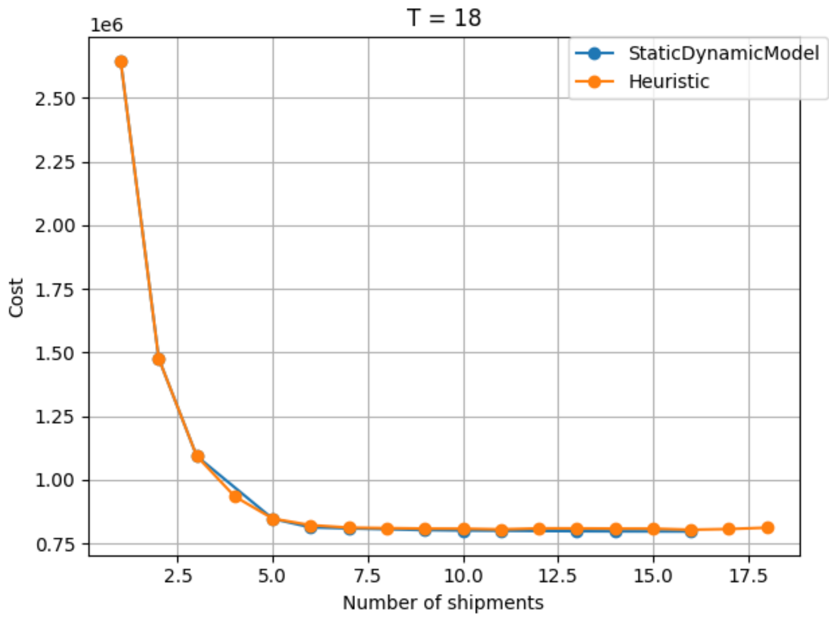

T = 6
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments

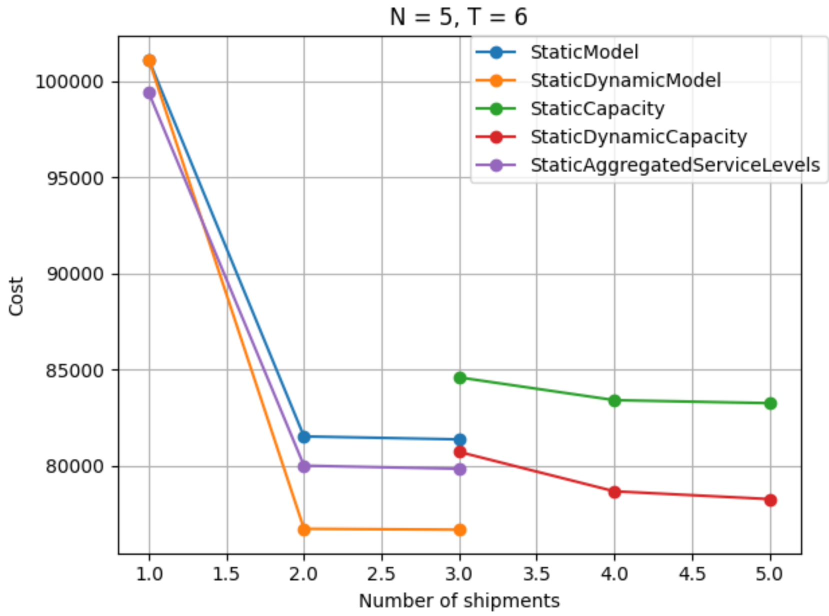

T = 6
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Chosen pro
Cost
Number of shipments

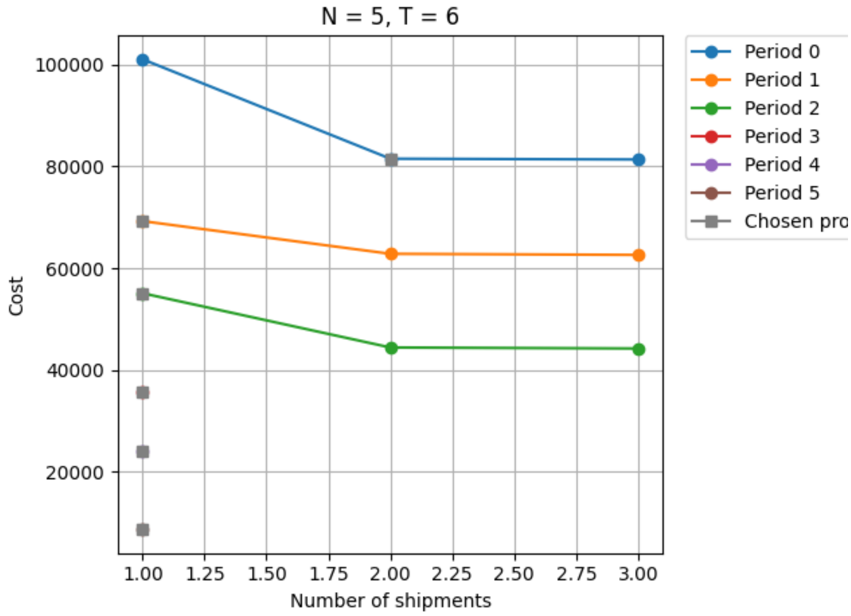

T = 6
StaticDynamicModel
Heuristic
Cost
Number of shipments

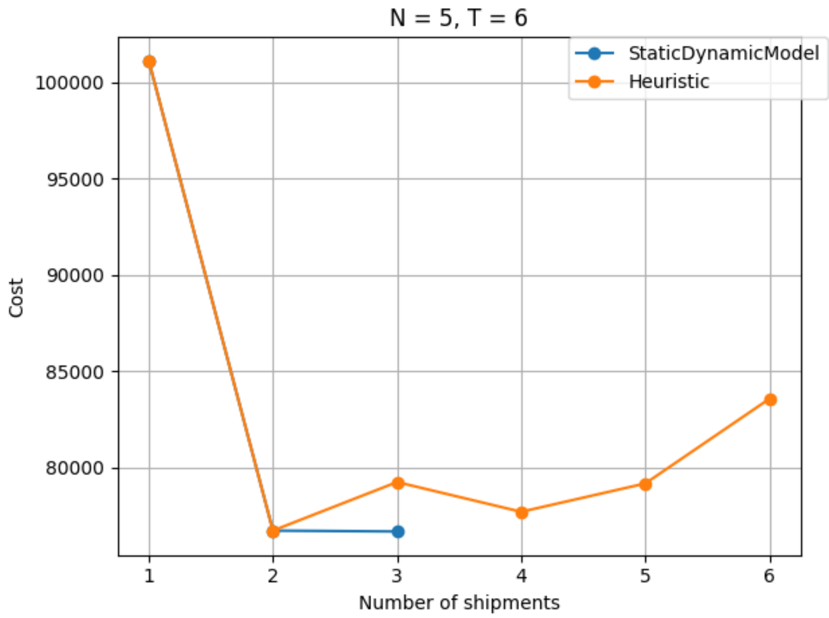

T = 18
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels

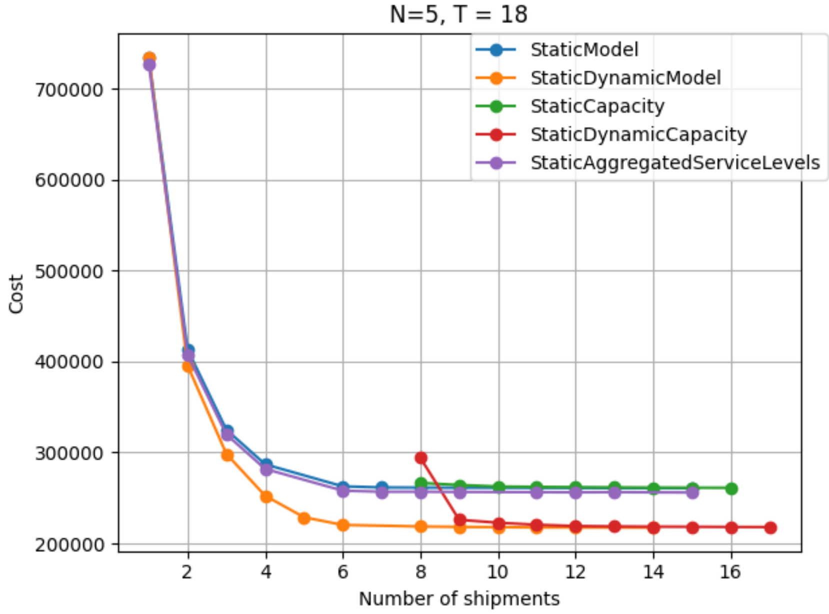

T = 18
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Period 12
Period 13
Period 14
Period 15
Period 16
Period 17
Chosen prod

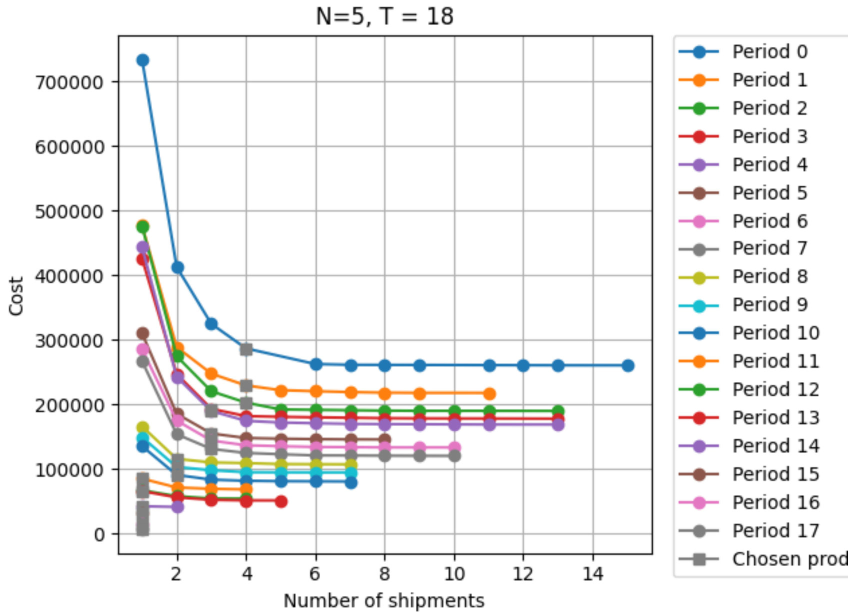

T = 18
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic

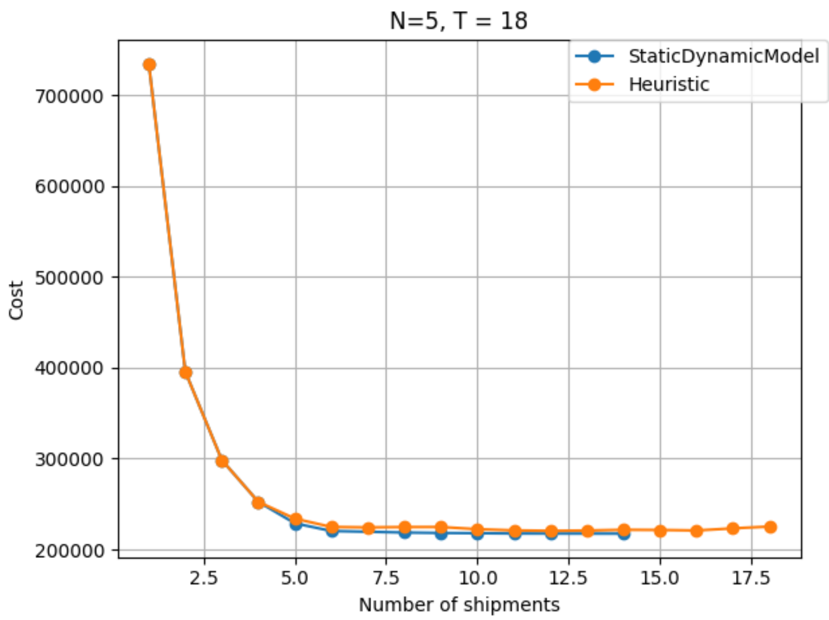

N = 5, T = 6
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments

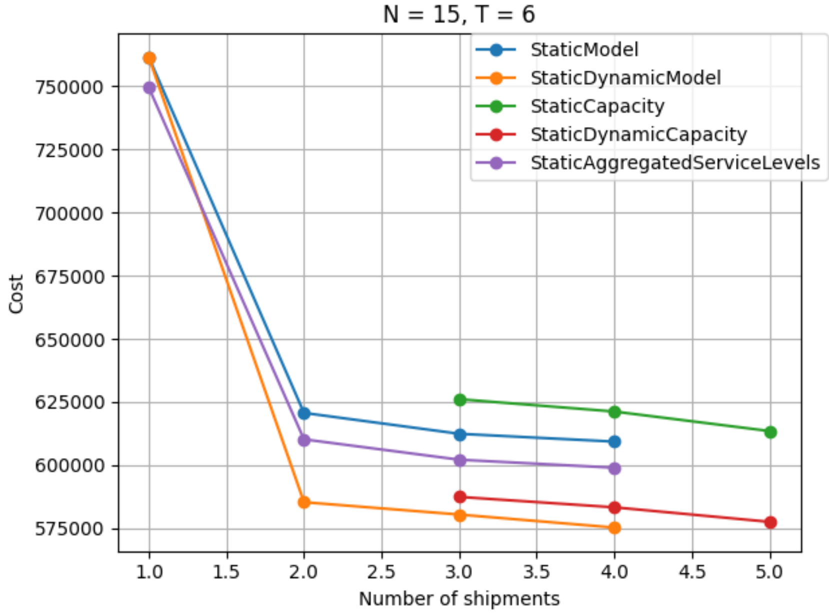

N = 5, T = 6
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Chosen pro
Cost
Number of shipments

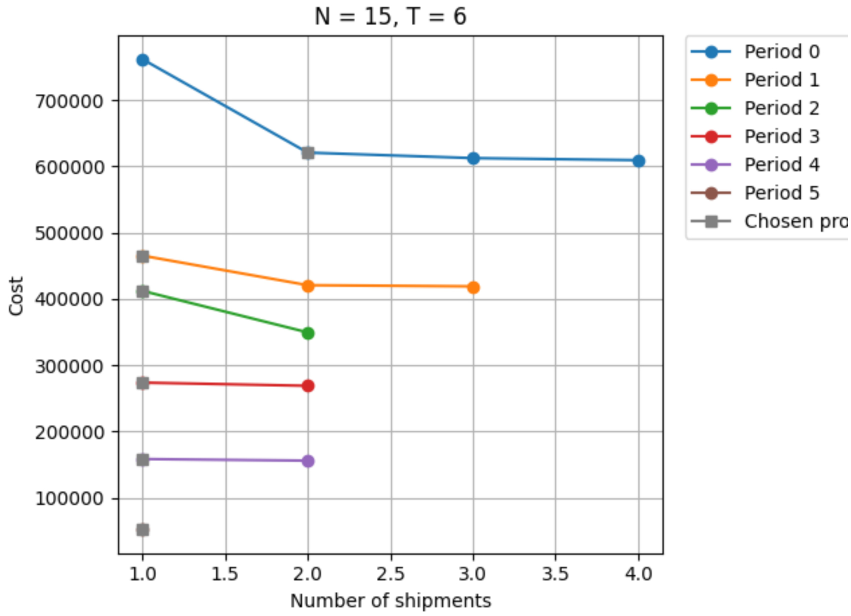

N = 5, T = 6
StaticDynamicModel
Heuristic
Cost
Number of shipments

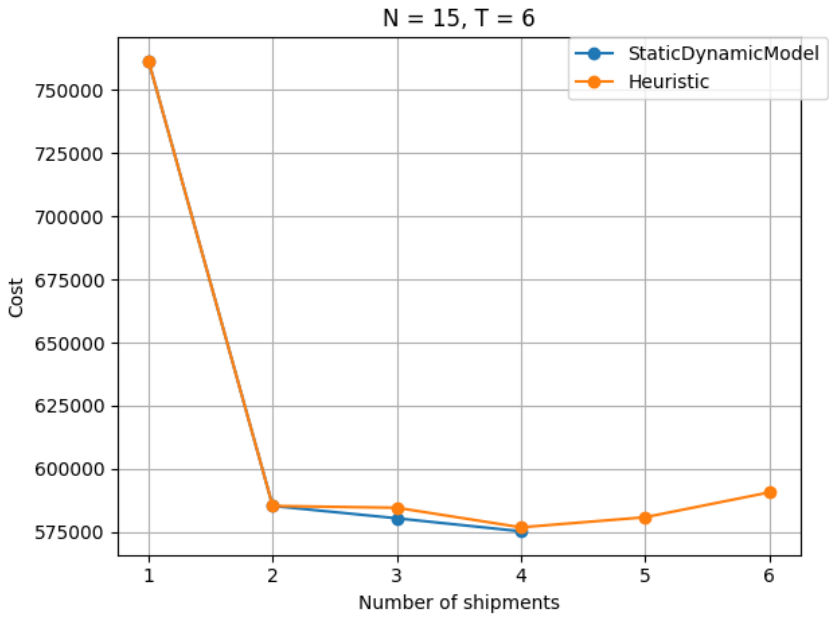

N=5, T = 18
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments
200000
300000
400000
500000
600000
700000
2
4
6
8
10
12
14
16

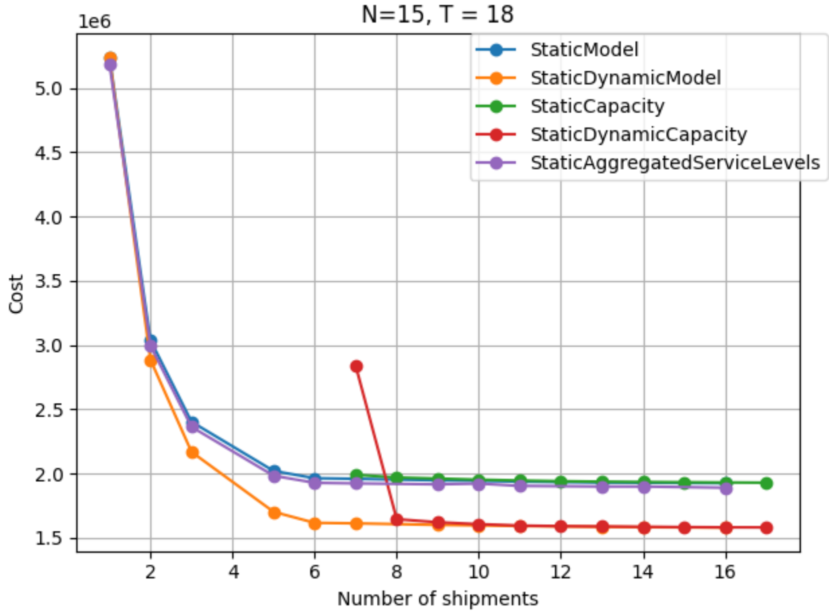

N=5, T = 18
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Period 12
Period 13
Period 14
Period 15
Period 16
Period 17
Chosen prod
Cost
Number of shipments
0
100000
200000
300000
400000
500000
600000
700000
2
4
6
8
10
12
14

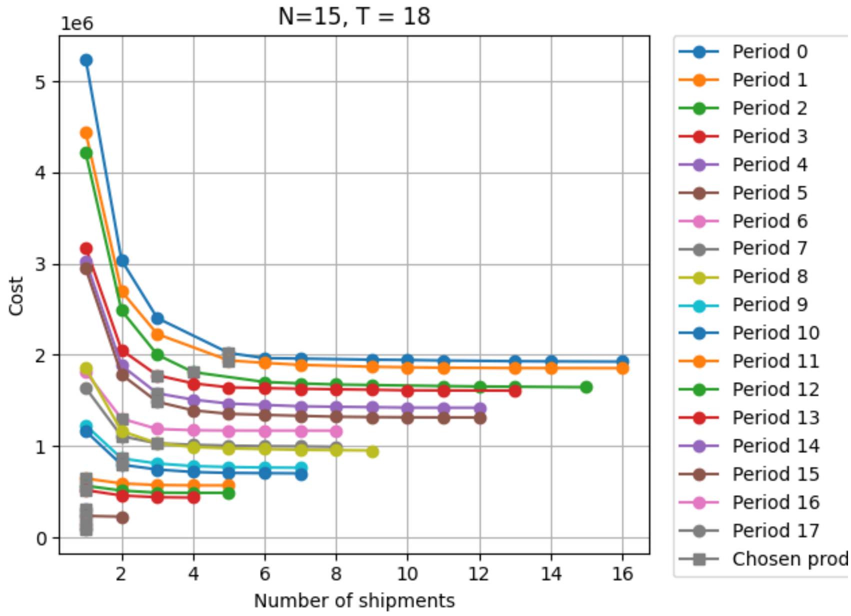

N=5, T = 18
StaticDynamicModel
Heuristic
Cost
Number of shipments
200000
300000
400000
500000
600000
700000
2.5
5.0
7.5
10.0
12.5
15.0
17.5

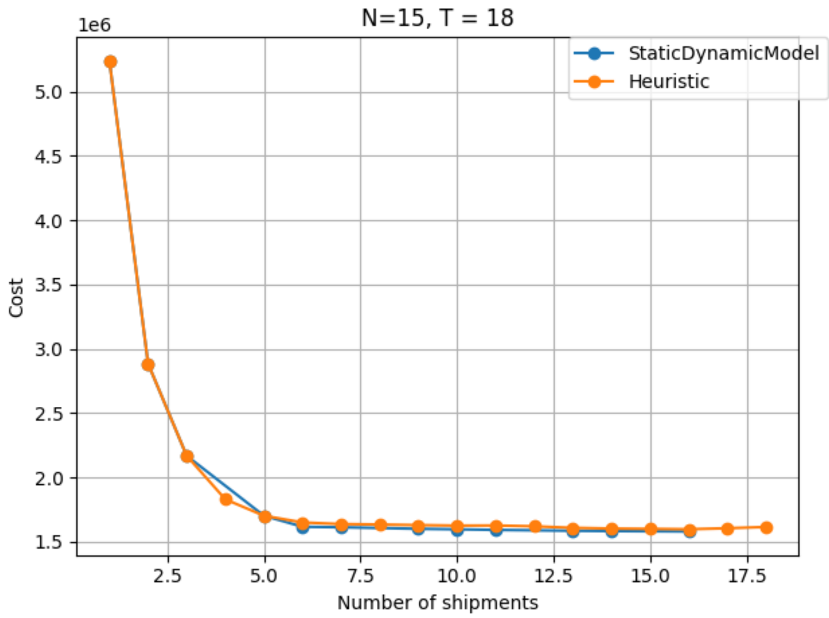

N = 15, T = 6
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments

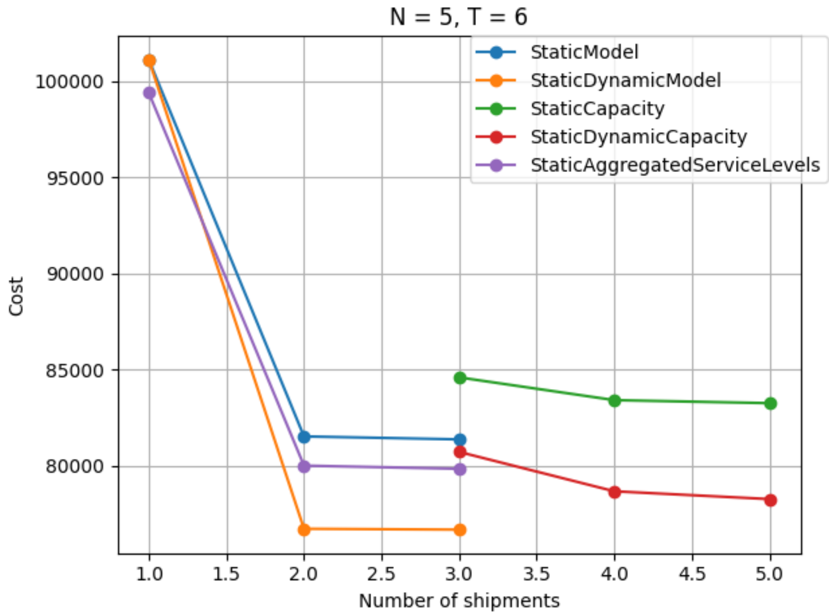

N = 15, T = 6
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Chosen pro
Cost
Number of shipments

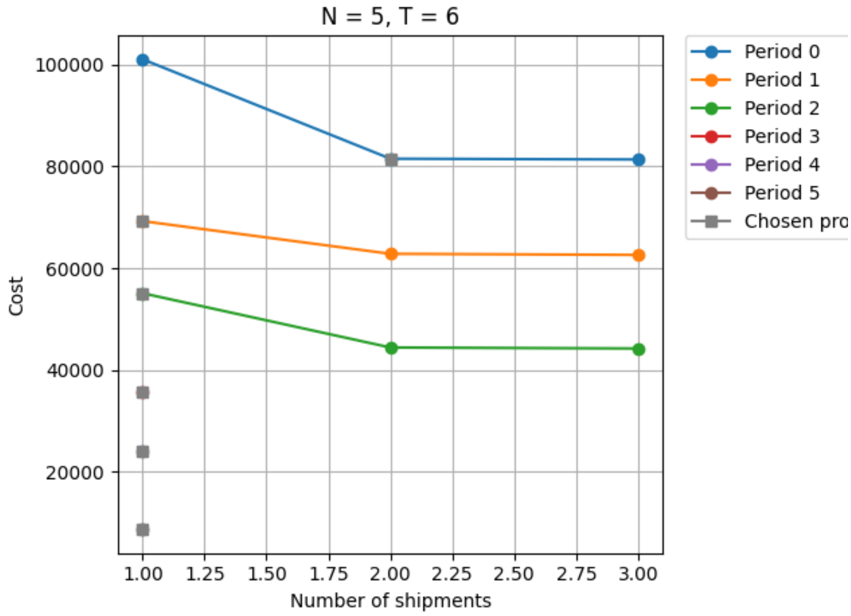

N = 15, T = 6
StaticDynamicModel
Heuristic
Cost
Number of shipments

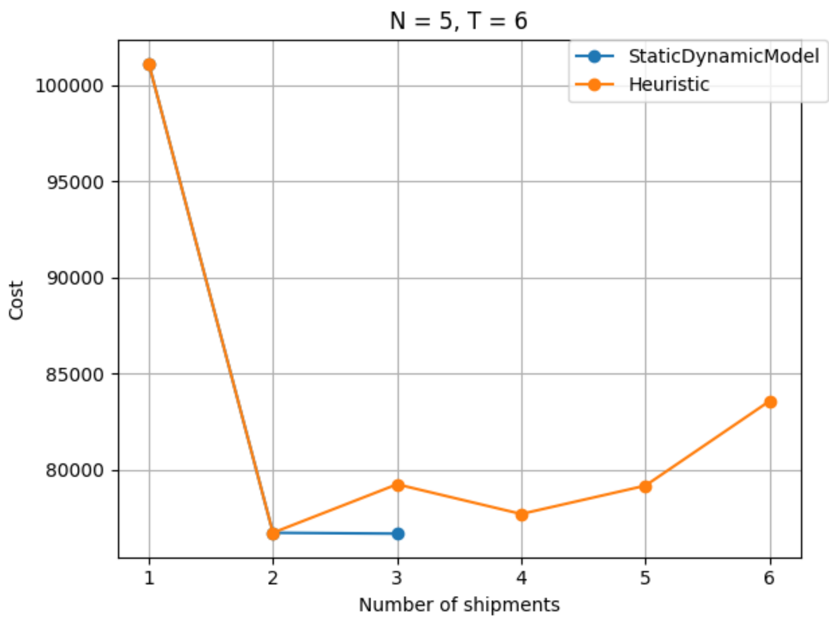

N=15, T = 18
1e6
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments

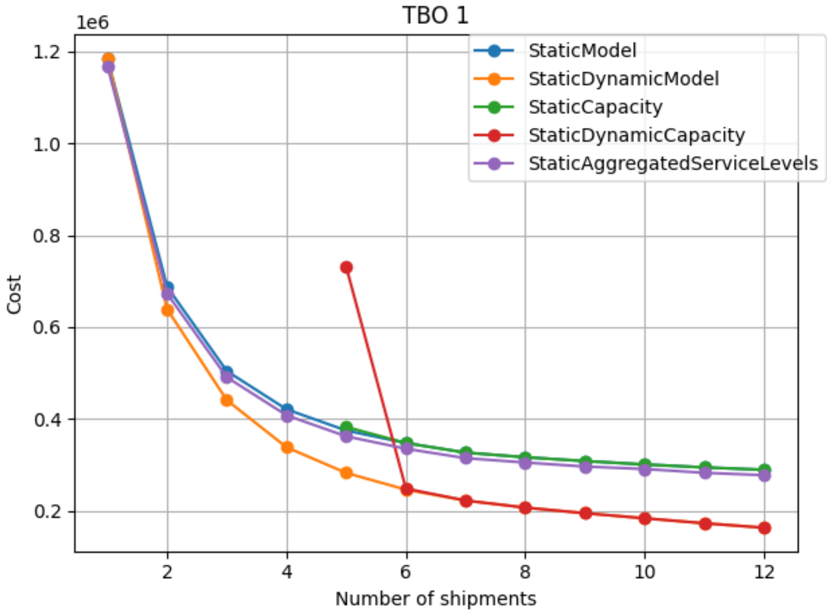

N=15, T = 18
1e6
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Period 12
Period 13
Period 14
Period 15
Period 16
Period 17
Chosen prod
Cost
Number of shipments

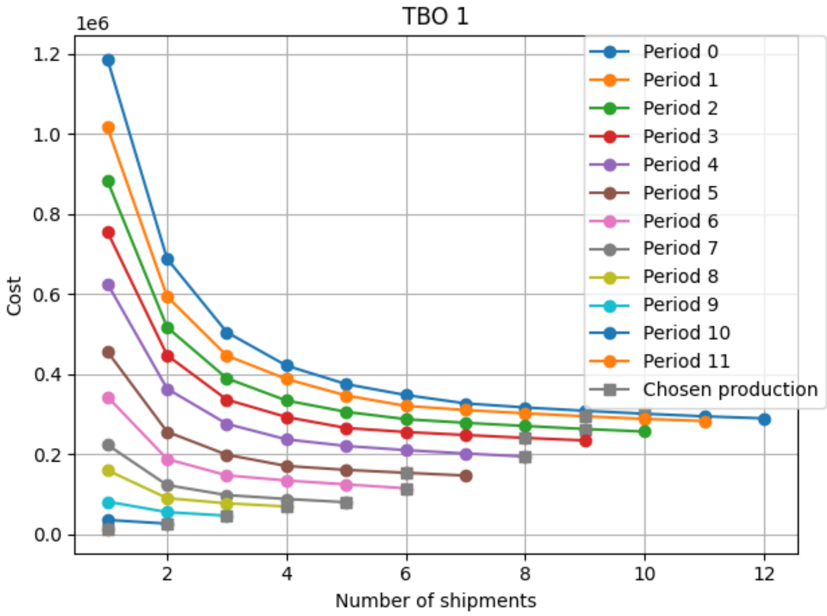

N=15, T = 18
1e6
StaticDynamicModel
Heuristic
Cost
Number of shipments

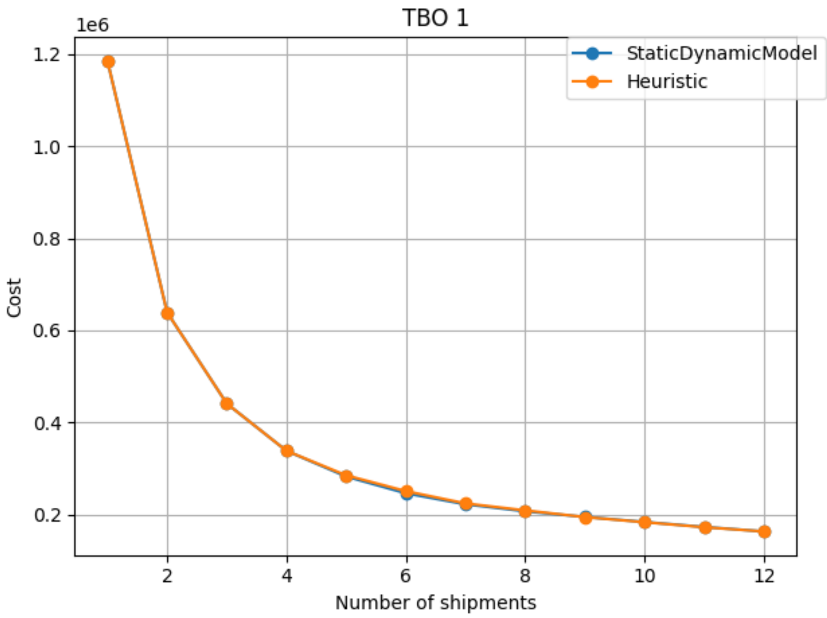

N = 5, T = 6
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments

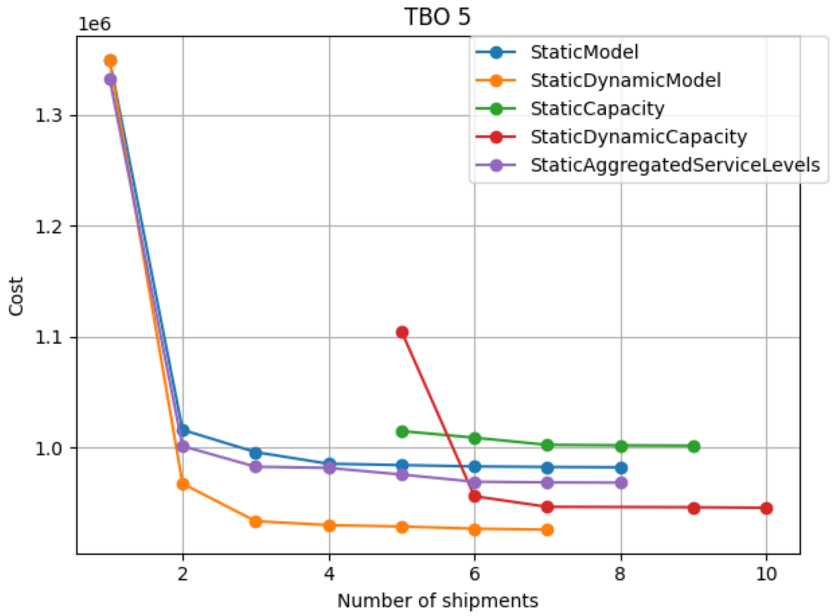

N = 5, T = 6
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Chosen pro
Cost
Number of shipments

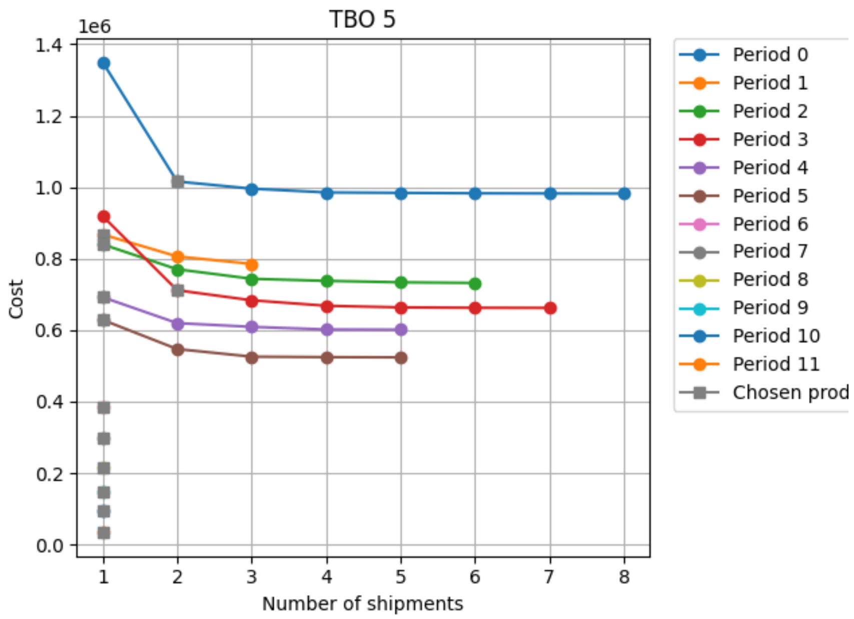

N = 5, T = 6
StaticDynamicModel
Heuristic
Cost
Number of shipments

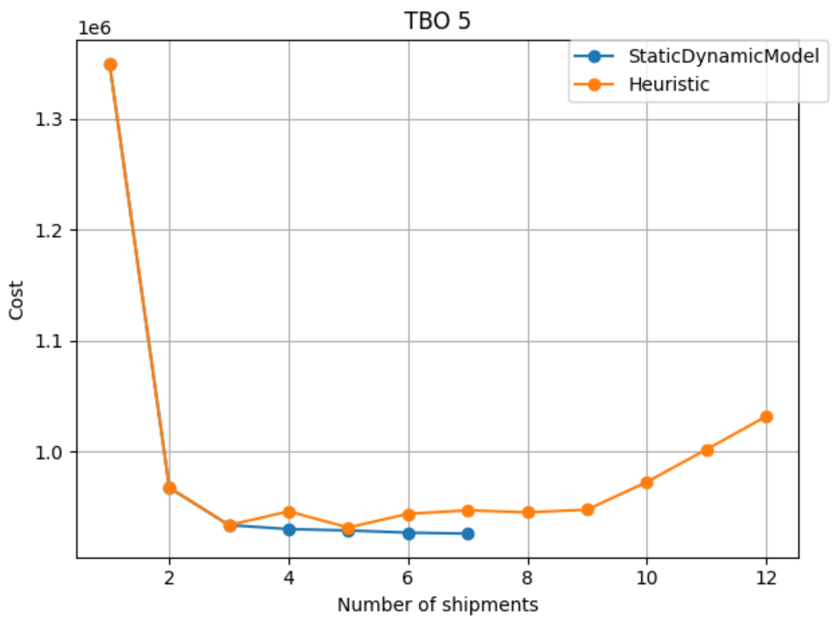

TBO 1
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels

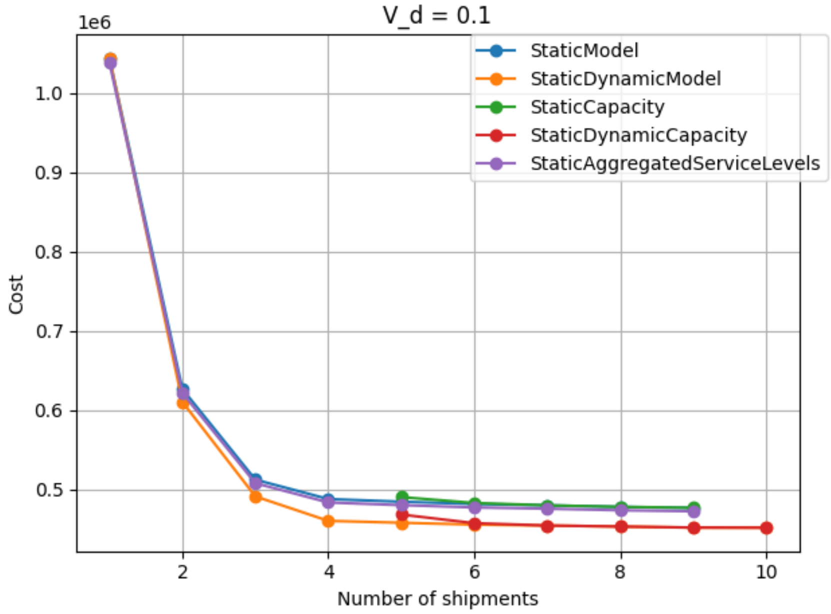

TBO 1
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen production

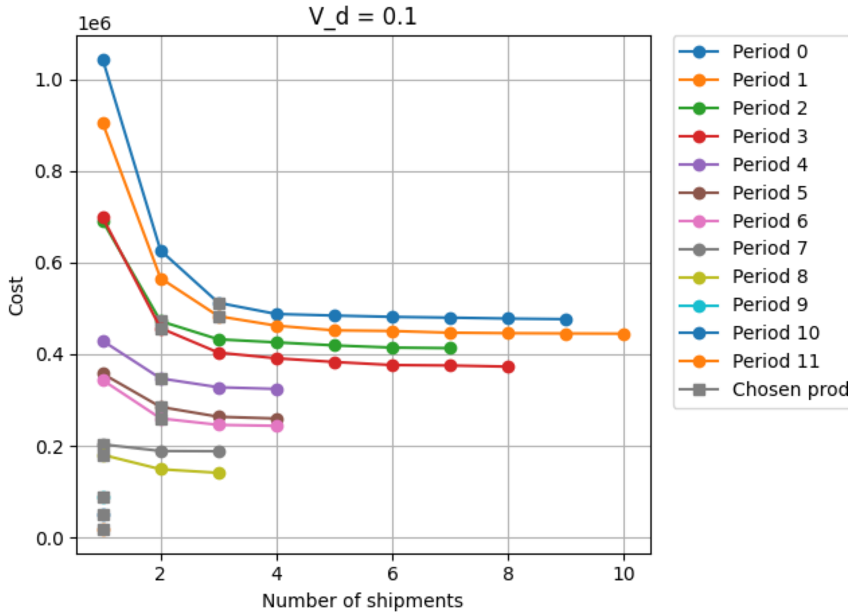

TBO 1
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic

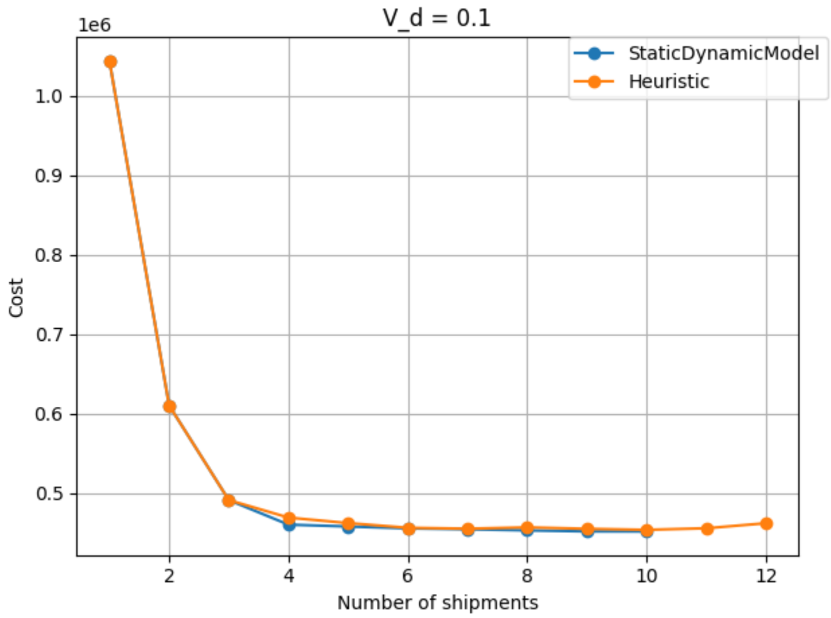

TBO 5
1e6
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels
Cost
Number of shipments

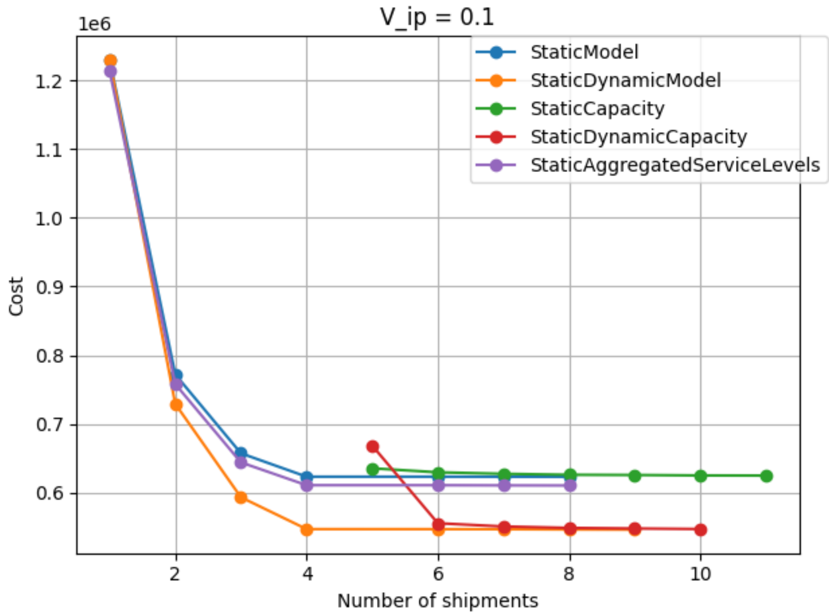

TBO 5
1e6
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod
Cost
Number of shipments


TBO 5
1e6
StaticDynamicModel
Heuristic
Cost
Number of shipments

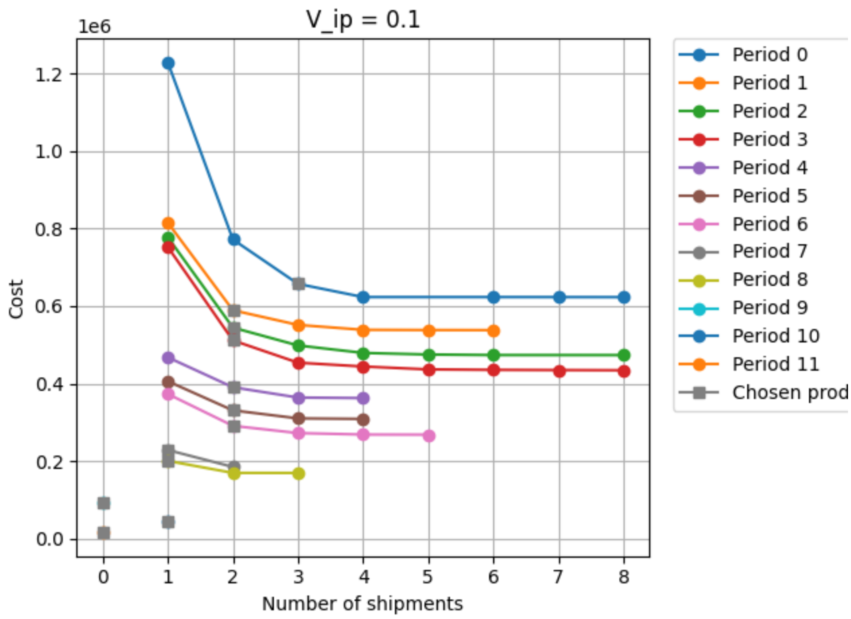

V_d = 0.1
1e6
Cost
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels


V_d = 0.1
1e6
Cost
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod

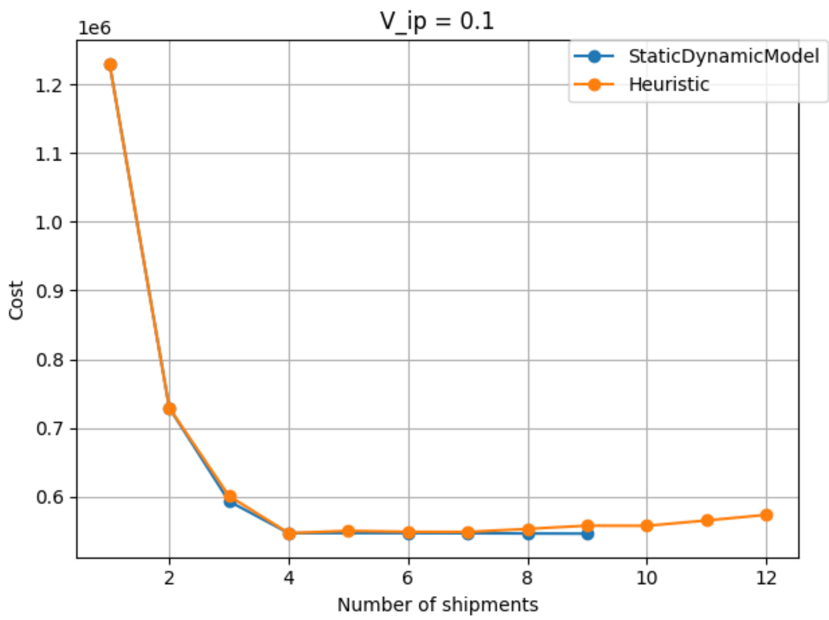

V_d = 0.1
1e6
Cost
Number of shipments
StaticDynamicModel
Heuristic



V_ip = 0.1
1e6
1.2
1.1
1.0
0.9
0.8
0.7
0.6
Cost
2
4
6
8
10
Number of shipments
StaticModel
StaticDynamicModel
StaticCapacity
StaticDynamicCapacity
StaticAggregatedServiceLevels



V_ip = 0.1
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Cost
0
1
2
3
4
5
6
7
8
Number of shipments
Period 0
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
Period 11
Chosen prod



V_ip = 0.1
1e6
1.2
1.1
1.0
0.9
0.8
0.7
0.6
Cost
2
4
6
8
10
12
Number of shipments
StaticDynamicModel
Heuristic