Integrated Forecast and Optimization for Retailer Allocation in a Two-Echelon Inventory System

Published Version:
Maniezzo, V., Zhou, T. (2023). Integrated Forecast and Optimization for Retailer Allocation in a Two-Echelon Inventory System. Cham : Springer [10.1007/978-3-031-42529-5_27].

Availability:
This version is available at: https://hdl.handle.net/11585/940950 since: 2023-09-08

Published:
DOI: http://doi.org/10.1007/978-3-031-42529-5_27

(Article begins on next page)

Integrated forecast and optimization for retailer allocation in a two-echelon inventory system

Vittorio Maniezzo[1] and Tingting Zhou[1,2]

[1] Department of Computer Science, University of Bologna, 47521 Cesena, Italy,
vittorio.maniezzo@unibo.it,
WWW home page: https://www.unibo.it/sitoweb/vittorio.maniezzo/en
[2] Department of Economics and Management, University of Science and Technology Beijing, Beijing, China
b20190402@xs.ustb.edu.cn

Abstract. The paper reports on a real-world application case of integrated data forecasting and process optimization for a tactical-level study in the context of the final stage of two-echelon logistics support for a large retail chain. The allocation of final stores to distribution centers had to be redefined in view of the expected increase in sales volume during the Christmas season. For this purpose, time series of past sales were projected up to the period of interest as a basis for the reoptimization of the store allocation. The latter problem was identified as an extended generalized assignment problem, which was solved using a Lagrangian matheuristic. Computational results are presented showing a comparison of different forecasting algorithms on the actual data and the advantages of using a matheuristic for optimization in this industrial setting, even when compared to results obtained by proprietary third-party solutions.

Keywords: Univariate time series forecasting, combinatorial optimization, Lagrangian decomposition, matheuristics

1 Introduction

Optimization is for the future. This is true for all business-related applications, and especially for logistics distribution, where goods must be efficiently delivered to destinations to satisfy future demand. This paper reports on a use case that calls for integrated forecasting and optimization for retailer allocation in a two-echelon inventory system [4], that is, in a two-level logistical distribution system for supply chain management that includes different Distribution Centers (DCs) and several retailers, each of which maintains its own inventory. The typical process supporting inventory management in this case is iterative, where at the beginning of each cycle, DCs order inventory from (external) suppliers and the orders arrive after a lead time. When the goods arrive, the DCs must have determined their allocation to each retailer. To reduce uncertainty, each time a DC makes an allocation decision, it typically relies on advance demand information and market forecasts, and may update future allocation settings.

A challenge for any economic system, including two-level distribution systems, is how to examine value and make the best use of this information. In our case, this also means that retailers should accurately quantify their next demand in the inventory management cycle and make appropriate inventory adjustments and operational planning when considering their internal store inventory planning, but the DCs themselves must have sufficient inventory to meet these forecasted demands. This paper reports on the problem of dimensioning multiple DCs and allocating stores to them to meet expected demand in a critical period.

The reported test-case consists of an actual two-echelon allocation instance where a distribution company needed to supply the 52 retail stores in a large city with goods forwarded from 4 different DCs. Specifically, the company operated (and still operates) a large DC on the outskirts of town, a smaller DC diametrically opposite the first, and two small satellites closer to the city center. These latter two DCs impose tight capacity constraints, but allow for less expensive and more timely service to most of the city's stores. Figure 1 shows the (anonymized) geolocation of the DCs and stores and the GIS model we developed.

In order to optimize the allocation of stores to DCs for the peak demand period of the Christmas season, we first forecast the aggregated demand expected by each store months in advance, and then solved the derived allocation problem instance using a Lagrangian matheuristic. The paper reports the results we obtained after these two phases.

The rest of the presentation is structured as follows. Section 2 contains a brief overview of the related literature on two echelon inventory systems, while section 3 details the elements of our test case. Section 4 presents the solution approach we followed, with special emphasis on the request forecasting module and on the allocation optimization module. Section 5 shows the computational results, both for the forecasting and for the optimization modules, and conclusions are drawn in section 6.

2 Related literature

The multi-echelon inventory concept was first introduced in [5], thereafter inventory control policies for multi-echelon systems have been widely studied in the organizational literature [22]. The rich literature on the various aspects of two-echelon inventory system optimization cannot be fully reported here, we only mention that various aspects have been studied beyond demand forecasting, including *inventory control policies*, i. e, policies for managing inventory in two-echelon systems that determine when and how much inventory should be replenished at each level based on factors such as demand forecasts, lead times, and desired service levels, *replenishment strategies*, strategies for coordinating replenishment decisions between the central warehouse and retail locations, considering factors such as batch ordering, cross-docking, and coordination of transportation activities, *risk management*, regarding uncertainties in demand, lead times, and other parameters that can significantly affect inventory performance.

Demand forecasting plays a central role in this area [21], and different forecasting algorithms have been proposed, ranging from simple moving average to sarima to combined approaches [23, 15, 16], showing how accurate demand forecasts can influence factor judgments by analyzing historical data characteristics and integrating them into management models [14][3]. For example, in [19], an improved autoregressive demand approach can achieve predictions of the correlation between time and retailer and help minimize the average expected system cost per period, or in [18], a forecast-driven tactical planning model for a serial manufacturing system attempts to simulate tradeoffs associated with production strategies through dynamic planning to achieve the correct balance between inventory and planned production to meet customer requirements.

As for optimization, a large number of mathematical models have been proposed [12, 7]. Most of these models assume stochastic or stationary external demand, and their objective is to establish an equation in terms of total system cost that determines the optimal lot size to be requested and the reorder point at each level of the supply chain. A mathematical model was developed in [2] for a two-level inventory system integrating a central warehouse with N different retailers, where the external demand was generalized from a Poisson to a compound Poisson demand, while [1] studied a two-level inventory system model with a central warehouse and N non-identical retailers. Subsequently, more recent developments introduced heuristic algorithm approaches for cost optimization, even when seasonal or cyclical demand is considered [20].

Effective multi-echelon inventory management is widely recognized for minimizing the average total cost of inventory by promoting coordination and cooperation among supply chain members. A decentralized inventory policy can in fact generate excess inventory without improving customer service levels [12]. Regarding the allocation of stores to DCs, to reduce uncertainty and associated potential costs, each time a DC makes an allocation decision, it can look for advance demand information and update its market forecast in anticipation of deriving the structural attributes of the model that provides management insight [18].

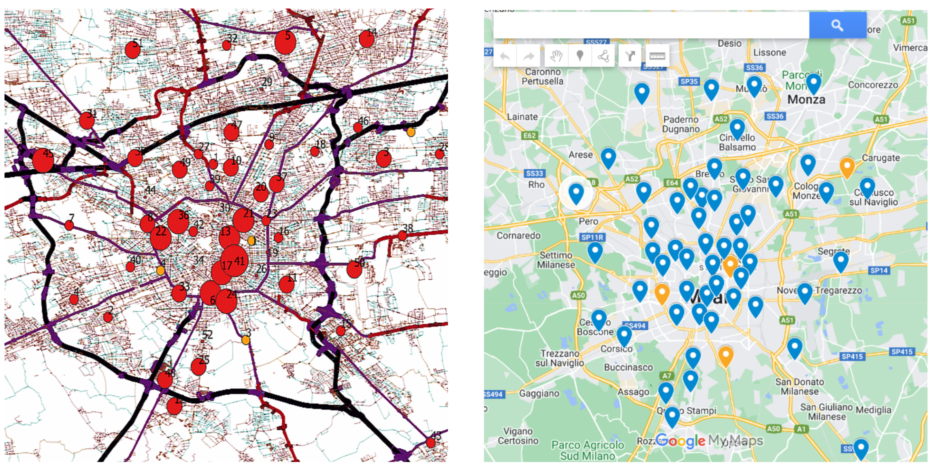
Most of the cases in the literature that report the implementation of multi-echelon inventory policies [10, 9] are based on examples of instances with generic demand simulated from multiple distributions. In the following, we report on a real-world use case where the demands are actual demand records.

3 Use case description

The problem to solve was defined at the tactical level, the request was to define the minimum cost distribution plan for the Christmas season, i.e., for December, defining the allocation of stores to DCs when the relevant orders were not yet available. The stores and DCs were already known, as were the capacities of the DCs (except for the largest one, which could have further space to rent but was presumably already oversized). The analysis was carried out excluding fresh products, which had their own line, and by aggregating orders and availabilities into “pallets”. The allocation scheme was later used as the basis for a more fine-grained inventory policy management, where “pallets” were broken down into specific product lines.

Each store's demand, which typically has a significant seasonal pattern, had to be independently forecast at least three months in advance, based on each customer's historical orders over the past 45 months. This period unfortunately included the months affected by the covid pandemic. The geolocation of stores and DCs allowed the cost of delivering a single pallet from each DC to each store to be quantified as a combination of road distance and travel time. Inventory levels at each DC were a decision variable, subject to DC capacity constraints.

Our system was to be used to support what-if analysis, i.e., to define quantitative scenarios that would later be refined to arrive at the final allocation plan. As such, it needed to provide a degree of flexibility, including the ability to fix some of the current allocations or to split specific customer requests between up to two DCs, even though only one of them would ultimately be selected to meet the customer's needs.



(a) GIS view (b) gmap view

Fig. 1: Input data

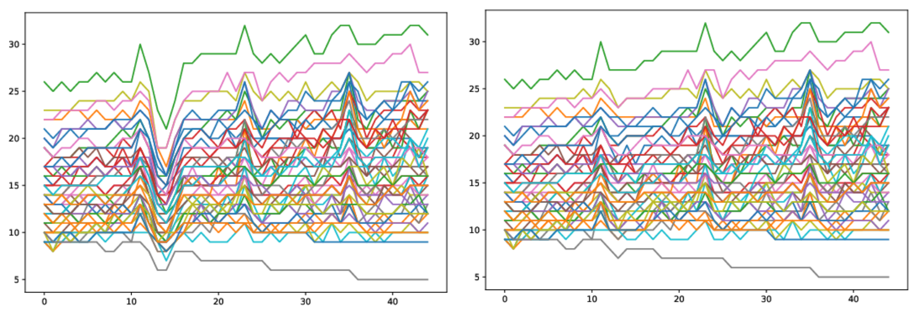
4 Methodology and system architecture

The complete solution required first to forecast the future demand of each of the stores of interest, and then to use the forecast results as input for allocating the stores to the DCs. The input data series were read and independently projected, and the results of the Christmas season forecasts were used as input to define the instance of the allocation problem, which was then solved using a Lagrangian matheuristic. The two activities, forecasting and optimization, could therefore be pipelined by delegating them to a forecasting module and an optimization module, respectively.

4.1 The forecasting module

The objective of the forecasting module was to determine the amount of goods requested by each store in the month of December by projecting the historical delivery time series from the last available data (month of August, thus including September orders) to the end of the year.

Historic data, besides trend and possibly seasonal components, was affected by the occasional drop in sales caused by the covid pandemic. This drop early in the series would affect the later forecasts. We fixed this in a preprocessing phase by regenerating the series values for the three most affected months (February, March and April 2020). This was done simply by computing the STL decomposition [6] on unaffected data, assuming a linear baseline, and computing and projecting the relevant seasonality coefficients on the three months to reconstruct. Results are shown in figures 2a and 2b.



(a) Actual data

(b) Preprocessed data

Fig. 2: Input data

Each store's time series was then tentatively extended by means of several alternative forecasting algorithms, where possible considering the seasonality indices obtained by the STL decomposition as an external predictor. The forecasting methods included SARIMAX, MLP, RNN (LSTM), random forest and gradient boosting (XGboost). The relevant hyperparameters were set by grid search when they were all integer-valued, such as in the case of SARIMAX, and by Hammersley sampling [13] when real valued parameters were involved.

4.2 The optimization module

Given the forecasts of prospective demand, the next step was to optimize the allocation of stores to DCs. The system was assumed to be used primarily to generate what-if reports that quantified a number of scenarios, including freezing most of the current allocations while leaving a few selected ones free, reoptimizing the entire set of allocations, and accepting some multiple (actually double) allocations to be refined at a later stage.

The problem has been modeled as an extended generalized assignment problem (GAP, [17]), where assignment constraints could be relaxed to accept up to a parametric number of assignments. Denotators for the actual application elements (stores and DCs) and for their models (clients and servers) will be used interchangeably. The notation used for the model is as follows:

n number of clients (stores)

J index set of clients, $J = \{1, \dots, n\}$
m number of servers (DCs)
I index set of servers, $I = \{1, \dots, m\}$
c_{ij} generalized cost for allocating client $j \in J$ to server $i \in I$
Q_i capacity of server $i \in I$
b_j maximum number of servers (DCs) client j can be assigned to
x_{ij} binary decision variable equal to 1 iff client j is allocated to server i, $i \in I$ and $j \in J$
q_{ij} integer decision variable representing the amount (number of pallets) of request of client j provided from server i

In the final test case, the amount requested by each client was always equal to the corresponding predicted value, but in some scenarios, the requests of some clients could be explicitly split between two servers, so the requested q_{ij} amounts could vary.

The mathematical model used for this problem was as follows.

$$z_P = \min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (1)$$

$$\textbf{subject to} \quad \sum_{j \in J} q_{ij} \leq Q_i \qquad i \in I \quad (2)$$

$$\sum_{i \in I} q_{ij} = req_j \qquad j \in J \quad (3)$$

$$\sum_{i \in I} x_{ij} \leq b_j \qquad j \in J \quad (4)$$

$$q_{ij} \leq req_j x_{ij} \qquad i \in I, j \in J \quad (5)$$

$$x_{ij} \in \{0, 1\} \qquad i \in I, j \in J \quad (6)$$

$$q_{ij} \in \mathbb{Z}_0^+ \qquad i \in I, j \in J \quad (7)$$

This model is an extension of the GAP in that the number of servers for client $j \in J$ can vary from 1 to b_j. In case of $b_j = 1$ problem P reduces to the standard GAP.

The test case was small enough to ensure a fast solution using commercial MIP solvers, but it was preferred not to rely on third-party software. The combination of the ability to quickly obtain an optimal solution and the desire for in-house software development led us to choose a matheuristic for the solution. Matheuristics [17] are heuristic algorithms that take advantage of mathematical programming (MP) modules to obtain possibly non-optimal solutions. However, their MP components allow to obtain strong guarantees, such as an optimality test, a quality bound, or possibly dual information to exploit during the search. Given the structure of the problem and the expected results, we chose a decomposition heuristic based on Lagrangian relaxation as the core problem optimizer. Lagrangian relaxation is a mathematical programming approach to constrained optimization that approximates the original problem by removing difficult constraints and incorporating ("dualizing") them into the objective function. See [17] for details.

One possible Lagrangian relaxation of problem P dualizes constraints 2 adding them to the objective function with penalties λ_i, $i \in I$. The relaxed formulation becomes:

$$z_{LR}(\lambda) = \mathbf{min} \sum_{i \in I} \sum_{j \in J} (c_{ij} x_{ij} + \lambda_i q_{ij}) - \sum_{i \in I} \lambda_i Q_i \tag{8}$$

$$\textbf{subject to } (3), (4), (5), (6), (7) \tag{9}$$

$$x_{ij} \leq q_{ij} \qquad i \in I, j \in J \tag{10}$$

$$\lambda_i \geq 0 \qquad i \in I \tag{11}$$

Notice the need to include constraints 10, which are redundant in formulation P but become necessary once variables q_{ij} get non null costs due to the penalty contribution.

Formulation LR includes an easy to solve Lagrangian subproblem, $LR(\lambda)$, whose solution can be included in a subgradient optimization routine [17] to solve the Lagrangian dual problem and obtain an effective penalty vector. The overall structure of the optimization code is presented in algorithm 1. Steps (5) and (8) are problem-specific and in this case can be based on the direct inspection of the penalized costs.

Algorithm 1: Core Lagrangian heuristic

1 procedure LagrHeuristic;
Input : Control parameters
Output: A feasible solution $\mathbf{x}^*$ of value z*
2 Initialize the penalty λ;
3 Identify the subproblem LR(λ);
4 **repeat**
5   solve subproblem LR(λ) and get the possibly infeasible solution $\mathbf{x}$;
6   check for unsatisfied constraints ;
7   update the penalty λ ;
8   construct problem solution $\mathbf{x^h}$ using $\mathbf{x}$ and λ ;
9   **if** $z(\mathbf{x^h}) < z^*$ **then** $\mathbf{x}^* = \mathbf{x^h}$; // $z(\mathbf{x^h})$, z^* are costs of $\mathbf{x^h}$, $\mathbf{x}^*$
10 **until** *end_ condition*;

5 Computational results

The algorithm 1 was implemented in Python 3.9 and ran on a Windows 10 machine with 16 GB of RAM. Neural models were implemented on tensorflow 2.8.0. For the purposes of the presentation in this paper, we also ran an independent test with CPLEX 20.1 on the data available at the end of the summer to validate the results.

Forecasting results
The 45 month datasets were decomposed into testsets of 42 items and validation sets of 3 items each, corresponding to the time span required to be forecasted. The algorithms used for forecasting were finally compared in terms of effectiveness, figure 3 shows the corresponding critical difference diagram [8, 11] based on the Wilcoxon-Holm signed-rank test, i.e. the pairwise post-hoc analysis based on the Wilcoxon signed-rank test with Bonferroni-Holm correction and $\alpha = 0.05$.

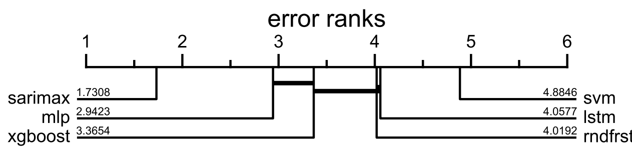


Fig. 3: Critical difference diagram

Figure 3 shows that on the 52 series of interest SARIMAX ranked significantly better than then the other 5 algorithms, then multilayer perceptrions (MLP) and xgboost were significantly equivalent, with MLP ranking a little better. Xgboost was on the other side equivalent to random forest, which was in turn equivalent to LSTM. Note that the tiny difference in ranking between LSTM and random forest was sufficient to make the former significantly different in ranking from xgboost. Finally, SVM (used as SVR) was an approach that on our series ranked significantly worse than all other algorithms. The descriptive statistics of the relevant data are reported in table 1 and substantiate the results of figure 3. The columns show the mean and the median error computed for validation over the 52 series, along with the standard deviation of the errors. The cited significance of the differences are computed by the mentioned Wilcoxon-Holm tests.

Table 1: Descriptive statistics on forecast errors

	mean	median	st.dev
sarimax	0.05	0.36	1.78
mlp	2.10	1.82	3.07
xgboost	2.54	2.00	2.08
rand.for.	2.70	2.12	2.10
lstm	2.87	2.71	2.32
svm	5.40	5.00	5.30

Optimization results
Figure 4 shows a trace of the evolution of the Lagrangian lower bound (z_{lb}) and the corresponding heuristic solution (z_{ub}) for evaluating a specific scenario. The upper and lower bound traces are superimposed on the optimal value line (*opt*) and both converge to it, and in fact the optimality of the solution can be determined by the heuristic itself, in this case with an uncharacteristic monotonic increase of the lower bound. This optimality guarantee was not achieved for all scenarios, but in all cases the difference between the upper and lower bounds at

the end of the search was less than 1%, and the final solution was obtained in less than 5 seconds of CPU time. It is noteworthy that in all the tests we performed, i.e., in all the scenarios we evaluated, the heuristic was able to produce the optimal solution, if not prove its optimality, as certified by the independent CPLEX results.

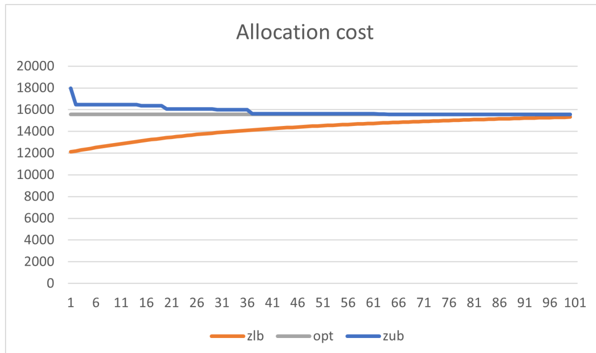


Fig. 4: Lagrangian heuristic, upper and lower bound evolution

6 Concluding remarks

The paper reports on a study of a use case for demand forecasting and allocation optimization for a two-echelon logistics system, where several DCs were to be used to supply goods to final retail stores scattered around a large city. The project required forecasting several months in advance, i.e. at a tactical level, the requests that would eventually be made by the stores to the DCs, and consequently optimizing their allocation. The forecasts were complicated by the occasional impact of the pandemic on the input data series.

Computational tests were performed on validation sets taken from the input series. Different forecasting algorithms were tested and compared, with the result that in our case statistical forecasting was significantly more reliable than alternative AI-based methods. The relatively simple derived allocation problem was solved by a Lagrangian matheuristic, actually considering several what-if scenarios derived from different constraints imposed on the desired solution. The code exhibited desirable properties specific to matheuristics in that, without the need for third-party libraries, it was able to produce the optimal solution for all scenarios and even prove its optimality for several of them.

References

1. J. Andersson and J. Marklund. Decentralized inventory control in a two-level distribution system. *European J. of Operational Research*, 127(3):483–506, 2000.
2. S. Axsäter. *Inventory Control*. Kluwer Academic Publishers, Boston, USA, 2000.
3. I. Biçer and R. W. Seifert. Optimal dynamic order scheduling under capacity constraints given demand-forecast evolution. *Production and Operations Management*, 26(12):2266–2286, 2017.
4. W.K. Chiang and G. E Monahan. Managing inventories in a two-echelon dual-channel supply chain. *European J. of Operational Research*, 162(2):325–341, 2005.

5. A.J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science*, 6(4):475–490, 1960.
6. R.B. Cleveland, W.S. Cleveland, J.E. McRae, and I.J. Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–33, 1990.
7. Z. Dai, F. Aqlan, and K. Gao. Optimizing multi-echelon inventory with three types of demand in supply chain. *Transportation Research Part E: Logistics and Transportation Review*, 107:141–177, 2017.
8. J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7(1):1–30, 2006.
9. G. Eppen and L. Schrage. Centralized ordering policies in a multi-warehouse system with lead times and random demand. In L. Schwarz, editor, *Multi-Level Production/Inventory Control Systems*, pages 51–69. North Holland, 1981.
10. J.P. Escorcia-Caballero, R. Amaya-Mier, M. Soto-Ferrari, and O. Chams-Anturi. Multi-echelon inventory management policies: A case study for a two-echelon supply chain. pages 165–175, 11 2020.
11. I.H. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33:917–963, 2019.
12. A. T. Gumus, A.F. Guneri, and F. Ulengin. A new methodology for multi-echelon inventory management in stochastic and neuro-fuzzy environments. *International Journal of Production Economics*, 128(1):248–260, November 2010.
13. J.M. Hammersley. Monte carlo methods for solving multivariable problems. *Annals of the New York Academy of Sciences*, 86(3):844–874, 1960.
14. J. Holmström, K. Främling, R. Kaipia, and J. Saranen. Collaborative planning forecasting and replenishment: new solutions needed for mass collaboration. *Supply Chain Management*, 7(3):136–145, 2002.
15. M. J. Keneley and B. Van Der Zwaan. Demand forecasting and inventory control: A zero-inflated approach. *European Journal of Operational Research*, 2002.
16. Z. Ma, H. Li, and Y. Wang. Combining forecasts in supply chain forecasting: A review. *Omega*, 2015.
17. V. Maniezzo, M.A. Boschetti, and T. Stützle. *Matheuristics, algorithms and implementations*. EURO Advanced Tutorials on Operational Research. Springer. Springer International Publishing, New York, 2021.
18. F. Papier. Supply allocation under sequential advance demand information. *Operations Research*, 64(2):341–361, 2016.
19. G. Refik. A two-echelon allocation model and the value of information under correlated forecasts and demands. *European J. of Operational Research*, 99(2):386–400, 1997.
20. N. Sakulsom and W. Tharmmaphornphilas. Heuristics for a periodic-review policy in a two-echelon inventory problem with seasonal demand. *Computers & Industrial Engineering*, 133:292–302, 2019.
21. A. G. Shenstone and R. Dekker. A literature review on approaches to the forecasting of intermittent demand. *Journal of the Operational Research Society*, 2001.
22. M. Subrata. Inventory management in a two-echelon closed-loop supply chain with correlated demands and returns. *Computers & Industrial Engineering*, 62(4):870–879, 2012.
23. A. A. Syntetos and J. E. Boylan. Demand forecasting and inventory control for intermittent demand using a weighted moving average approach. *International Journal of Production Economics*, 2005.