Data Driven Decision Making for an Inventory Problem: A Rolling Horizon Multistage Stochastic Programming Approach

Rahman Khorrmfar, Saeed Chavoshi

rkhorra@ncsu.edu, schavos@ncsu.edu

October 2019

1 Introduction

This report summarizes the authors' effort to tackle the problem described in the contest. This INFORMS Competition, or as it dubbed as *A Contest fORged by Amazon Web Services and cORe* by the organizers, encourages participants to develop a *data-driven* approach that best accommodates the reproducibility of the computational experiments. The problem is an inventory management problem for which the data of the past 10 years is available. The data is provided to get insight and manage the system for the upcoming 2 years (24 months) in the best possible way. As a practical problem, every good methodology should leverage the historical data and be problem specific. A good methodology is the one that is grounded in the problem setting and produces an implementable framework within which the problem is handled properly. The definition of a good methodology implies that every approach that requires unreasonable amount of time to obtain quality solution is ruled out.

There are potentially several optimization methods to tackle the inventory problem. Dynamic programming and mathematical formulation are procedures proven to be very effective in many optimization problems, especially those related to inventory systems. Given the uncertain nature of the problem, each of these procedures could be equipped with different paradigms to handle demand uncertainty. Stochasticity sometimes is handled by assuming probability distribution for the uncertain data. Combining this assumption with the two procedure yields Stochastic Programming (SP) and Approximate Dynamic Program (ADP). Stochastic programming basically seeks a solution that is feasible for all possible realizations of parameters (called scenarios) over a finite number of periods. It is usually applied to settings where decisions are made repeatedly under the same circumstances [7]. This approach is reasonable to use when historical data is available [7] since the probability distribution can be estimated. However, the size of the stochastic problem could get intractable when the outcome of each period and the number of periods increase. ADP is another effective way to handle stochasticity. It makes sequential decisions each using an estimate of the value of states [4]. Although ADP has been proved to be effective in certain problems, it is prone to the so called curse of dimensionality. ADP easily gets crippled when the state set, which is the time periods in our problem, and action set, which is the ordering quantity here grow. With the availability of the historical data and periodic nature of the problem, it seems that the SP with some considerations and adjustments, is a suitable choice for the purpose of this contest.

We propose a dynamic inventory management approach that simultaneously obtains insights from the historical data and considers the intrinsic uncertainty in the future demand. We apply a tailored forecasting method to utilize the information revealed so far and use multistage stochastic program to hedge against the future demand uncertainty. This approach ties forecasting and the decision making under uncertainty on a rolling horizon basis. Rolling horizon procedure (RHP) is a common industry practice for dynamic stochastic environments [6], especially in production planning. RHP iteratively solves a series of linked stationary problems over a known time horizon for which the demand estimation is reliable [5]. After demand realization for the next time period, RHP solves the next stationary problem with updated data. We formulate each stationary problem as a type of multistage stochastic Uncapacitated Lot-sizing Problem

(ULP). ULP optimizes production and inventory holding costs to satisfy future demand over a finite discrete-time planning horizon [2]. In this problem; however, we consider order amount as the production quantity in the ULP. Our methodology has three major steps: 1) Forecasting the next demand point 2) Generating the scenario tree, and 3) Combining the first two components by a RHP. Each of these components will be elaborated in the following sections.

2 Scenario Tree Generation

We already argued that mutlistage stochastic ULP would be a good choice. However, every mutlistage stochastic program is optimized over a scenario tree where demand and probability of each node is known. In essence, scenario tree approximately discretizes the demand uncertainty over a number of periods. The approximation arises since it contains a number of possible outcomes in each period. The size and structure of the scenario tree greatly impacts the SP model. Therefore, a trade-off between the level of discretization (number of outcomes) and demand approximation is necessary. Our preliminary results showed that a 4 period scenario tree with no more than 100 nodes offers a good balance between hedging against uncertainty and the total running time. Our methodology provides a good balance between the speed of the proposed framework and capturing demand uncertainty. We denote our tree by [1, o_2, o_3, o_4] where $o_2 - o_4$ are outcomes of each node at the previous period. Fig. 1 illustrates an example of a [1, 5, 3, 3] tree. The figure also depicts scenario and their corresponding probabilities. Each scenario is defined as a unique path from the root node (node 0) to a leaf node.

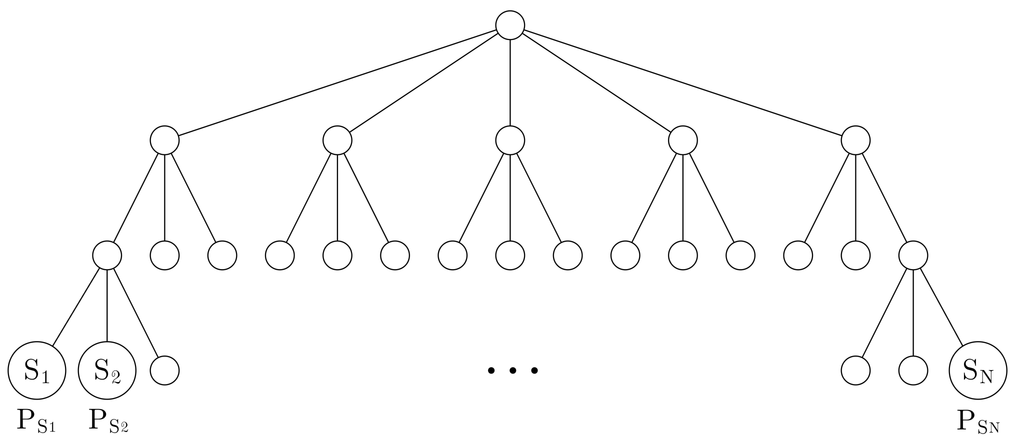

Figure 1: A [1, 5, 3, 3] scenario tree

Scenario tree generations requires a mechanism that systematically estimates demand and assigns a probability for each node. We apply the method proposed in [1] to carry out these two steps. The method is data-driven which matches the properties (moments) and distribution of demands to that of generated by the tree. It aptly combines the historical data with the current demand forecast [1,3]. This method is specifically suitable for inventory and production planning problems where the historical data is available (see examples in [1]). More specifically, we employed the linear programming version of the L_1-distribution matching method (or LP-DMP henceforth). This version is carried out in two steps where the first one sets demands, and probabilities are labeled in the second step.

2.1 Demand Forecasting

The overall process for estimating demand for each node is similar to the one proposed in [1]. The process starts with the root node and perform a one-step-ahead forecast for the next demand (base node). The generated demand is assigned to the middlemost child of the present node. Additional nodes are created by adding and subtracting multiples of the standard deviation of the base node's demand. Note that the structure of the scenario tree is set *a priori*, thus the number of children for each node is known. A process

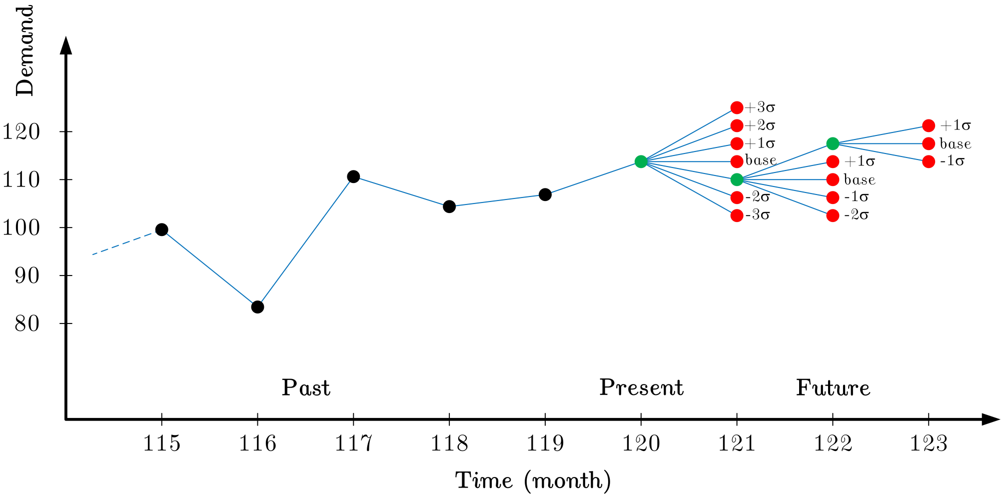

Figure 2: Demand forecasting for a [1, 7, 5, 3] scenario tree

of demand generation for a [1, 7, 5, 3] tree is depicted in Fig. 2. The process starts at the last available demand at period 120, which is the first planning horizon in our model. The base demand for the next period is estimated. Then, demand for other 6 children of node 0 is obtained by the base demand. The same process applied for each of the 7 nodes in period 1 (month 121 in the figure) with the updated demand. For instance, nodes in period 2 (month 122) are generated by available demand up to month 122 (including green nodes in the original data). The process continues until the entire tree is generated.

To perform the one-step-ahead forecast, We use Seasonal Auto-regressive Integrated Moving-Average (SARIMA). This method is a common forecasting method for univariate time series with potential trend and seasonality [3]. We tuned the hyper-parameters of SARIMA by an exhaustive grid search. The optimal values are obtained for trend parameters as (1, 0, 2) for *trend auto-regression order*, *trend difference order*, and *trend moving average order*, respectively. However, the tuning showed no specific trend pattern. Seasonality parameters are obtained as (2, 0, 1, 12) for *seasonal auto-regressive order*, *seasonal difference order*, *seasonal moving average order*, and *seasonality period*, respectively.

2.2 Distribution Matching Linear Program (DMLP)

Once the demand associated with each node of the scenario tree is known, a single optimization problem can compute probabilities (see [1] for in-depth treatment of the method). The entire scenario tree can be viewed as a collection of two-step sub-trees with every non-leaf node as a root and its children as the second stage outcomes. Let D_j be the estimated demand for the outcome j. The moment calculated from the data denoted as $\mathcal{M}_j$. We consider first four moments namely mean, variance, skewness, and kurtosis. Since L1 norm minimizes an absolute value function, we define m^+ and m^- with control parameters w_k to represent the positive and negative part of the moments calculated from the tree. Positive and negative deviation with respect to the Empirical Cumulative Distribution Function (ECDF) are denoted by θ^+ and θ^- with control parameter ω_j. Finally, lb is the lower bound of probabilities, and $\Phi(x)$ is the CDF of the standard normal distribution. For each two-stage SP the following problem give the probability for the child nodes.

$$
\begin{aligned}
\text{minimize} \quad & z = \sum_{k \in K \setminus 1} w_k (m_k^+ - m_k^-) + \sum_{j=1}^{N} \omega_j (\theta_j^+ - \theta_j^-) && \text{(1a)} \\
\text{subject to} \quad & \sum_{j=1}^{N} p_j = 1 && \text{(1b)} \\
& \sum_{j=1}^{N} D_j p_j = \mathcal{M}_1 && \text{(1c)} \\
& \sum_{j=1}^{N} (D_j - \mathcal{M}_1)^k p_j + m_k^+ - m_k^- = \mathcal{M}_k \qquad \forall k > 1 && \text{(1d)} \\
& \Phi\Big(\frac{D_j - \mathcal{M}_1}{\sqrt{\mathcal{M}_2}}\Big) - \sum_{j'=1}^{N} p_{j'} = \theta_j^+ - \theta_j^- \qquad \forall j = 1, ..., N && \text{(1e)} \\
& p_j \geq lb \qquad \forall j = 1, ..., N && \text{(1f)} \\
& m_k^+, m_k^-, \theta_j^+, \theta_j^- \geq 0, p_j \in [0, 1] && \text{(1g)}
\end{aligned}
$$

where the weighted absolute deviations between the statistical properties calculated from the tree and inferred from the forecast demand are minimized in (1a). (1b) ensures that the probabilities add up to 1. The mean of demands are matched in (1c). Higher order moments are matched in (1d). The deviation with respect to the actual CDF is captured in (1e). In (1f) we impose a lower bound for each probability to prevent L1 norm setting variables to 0. Finally, the type of variables are specified in (1g). Generation of the multistage scenario tree is carried out in a step forward manner as follows: 1) starting from the first period, solve DMLP for each non-leaf node. 2) multiply the probability of each child node to that of its parent node.

3 Model Formulation

In classical lot-sizing, the incurred costs (holding and back-order) are constants, and demand is deterministic for all periods. However, in this problem the demand is unknown which can be estimated at best and the holding cost is piece-wise linear. As mentioned, in our problem the order quantity bears the same concept of production quantity in the ULP. Given a multistage scenario tree with known demands and probabilities in each node, this section develops a mathematical model for the stochastic Uncapacitated Lot-sizing Problem, The main assumption for both deterministic and stochastic model are as follows:

- Demand is fulfilled at the beginning of each month.
- Lead time is 1 month for the ordered quantity.
- In the start of the planning (end of month 120), inventory level is 73. This inventory is carried out to he next period. Therefore, the system has at least incurred by the holding cost of these initial inventory.
- The order for the month $t + 1$ is placed at the beginning of month t. Received order for month $t + 1$ is used to update the inventory position at the same month.

First two assumptions are explicitly mentioned in the problem description, while the last two are inferred.

3.1 Deterministic Model

We start by defining sets, parameters and decision variables. Let T be the set of periods in the planning horizon. Let D_t denotes the demand of period t. Holding cost is 1 when inventory level is below 90, and 2 otherwise; $h_1 = 1$ represents the former case and h_2 for the latter. The back-order cost is defined as $b = 3$. Let decision variable Q_t be order quantity at the beginning of period t. I_t is the inventory level at the end of period t, and back-order for period t is B_t. The mathematical model for the deterministic ULP is as follows.

$$
\begin{array}{lllr}
ULP & \text{minimize} & \sum_{t\in T} h_1 \mathbb{1}(I_t \leq 90) + h_2 \mathbb{1}(I_t > 90) + bB_t & \\
& \text{subject to} & I_{t+1} = I_t + Q_t - D_{t+1} + B_{t+1} - B_t \qquad \forall t \in T & \text{(2a)} \\
& & Q_t \in \mathbb{Z}^+ \cup \{0\} \qquad \forall t \in T & \text{(2b)} \\
& & I_t \geq 0 \qquad \forall t \in T & \text{(2c)} \\
& & B_t \geq 0 \qquad \forall t \in T & \text{(2d)}
\end{array}
$$

The objective function minimizes inventory holding and back-order costs over the planning horizon. Constraint (4g) is the inventory balance constraint. Constraint (2b) to (2d) defines the variables. This model in the current form is non-linear due to indicator function in the objective. We define $Z_t = h_1 \mathbb{1}(I_t \leq 90) + h_2 \mathbb{1}(I_t > 90)$ which then the objective function becomes $\sum_{t\in T} Z_t$. Replacing $h_1 = 1$ and $h_2 = 2$, and with simple manipulation Z_t could be defined as $Z_t = (2 - Y_t)I_t = 2I_t - Y_t I_t$ where $Y_t = 1$ if $I_t \leq 90$, and $Y_t = 0$ otherwise. Let $W_t = Y_t I_t$; linearization techniques is readily available for W_t as it is a nonlinear term with a binary and a continuous variable. Linearization of the nonlinear terms yields the next model. Constraints (4a) to (4b) imposes if-then nature of the Y_t and linearizes the nonlinear terms.

$$
\begin{array}{lllr}
(LP-ULP) & \text{minimize} & \sum_{t\in T} 2I_t - W_t + 3B_t & \\
& \text{subject to} & I_t \geq (90+\epsilon)(1-Y_t) \qquad \forall t \in T & \text{(3a)} \\
& & I_t \leq 90 + M(1-Y_t) \qquad \forall t \in T & \text{(3b)} \\
& & W_t \leq MY_t \qquad \forall t \in T & \text{(3c)} \\
& & W_t \geq 0 \qquad \forall t \in T & \text{(3d)} \\
& & W_t \leq I_t \qquad \forall t \in T & \text{(3e)} \\
& & W_t \geq I_t - M(1-Y_t) \qquad \forall t \in T & \text{(3f)} \\
& & I_{t+1} = I_t + Q_t - D_{t+1} + B_{t+1} - B_t \qquad \forall t \in T & \text{(3g)} \\
& & Q_t \in \mathbb{Z}^+ \cup \{0\} \qquad \forall t \in T & \text{(3h)} \\
& & I_t \geq 0 \qquad \forall t \in T & \text{(3i)} \\
& & B_t \geq 0 \qquad \forall t \in T & \text{(3j)}
\end{array}
$$

3.2 Stochastic Lot-sizing Model

The model for stochastic uncapacitated lot sizing program (SULP) is proposed based on the LP-ULP. The notation and sets, parameters and variables are the updated for each scenario in the multistage scenario tree. P_s is the probability of each scenario node.

$$
\begin{array}{lllr}
(SULP) & \text{minimize} & \sum_{t\in T}\sum_{s\in\mathcal{S}} P_s(2I_t^s - W_t^s + 3B_t^s) & \\
& \text{subject to} & I_t^s \geq (90+\epsilon)(1-Y_t^s) \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4a)} \\
& & I_t^s \leq 90 + M(1-Y_t^s) \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4b)} \\
& & W_t^s \leq MY_t^s \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4c)} \\
& & W_t^s \geq 0 \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4d)} \\
& & W_t^s \leq I_t^s \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4e)} \\
& & W_t^s \geq I_t^s - M(1-Y_t^s) \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4f)} \\
& & I_{t+1}^s = I_t^s + Q_t^s - D_{t+1}^s + B_{t+1}^s - B_t^s \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4g)} \\
& & Q_t^s \in \mathbb{Z}^+ \cup \{0\} \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4h)} \\
& & I_t^s \geq 0 \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4i)} \\
& & B_t^s \geq 0 \qquad \forall t \in T,\ \forall s \in \mathcal{S} & \text{(4j)} \\
& & Q, I, Y, W, B \in \mathcal{N}_{\mathcal{S}} & \text{(4k)}
\end{array}
$$

4 Methodology

We already presented our approach for scenario tree generation and model formulation. Our rolling horizon methodology is depicted in Fig. 3 with a [1, 3, 3] tree (a small tree is chosen for the sake of figure clarity). It starts from the first planning horizon (month 120) and continues for 24 periods (one shift of horizon for each month). In each step, a scenario is generated and a SULP is solved. The solution of SULP gives the order quantity in the current period, which will be available in the next one. System parameters are updated (inventory level, backorder amount, holding cost, backorder cost) once the demand is realized for the next period. The next scenario tree is generated from the realized demand. This process continues until the end of the planning horizon.

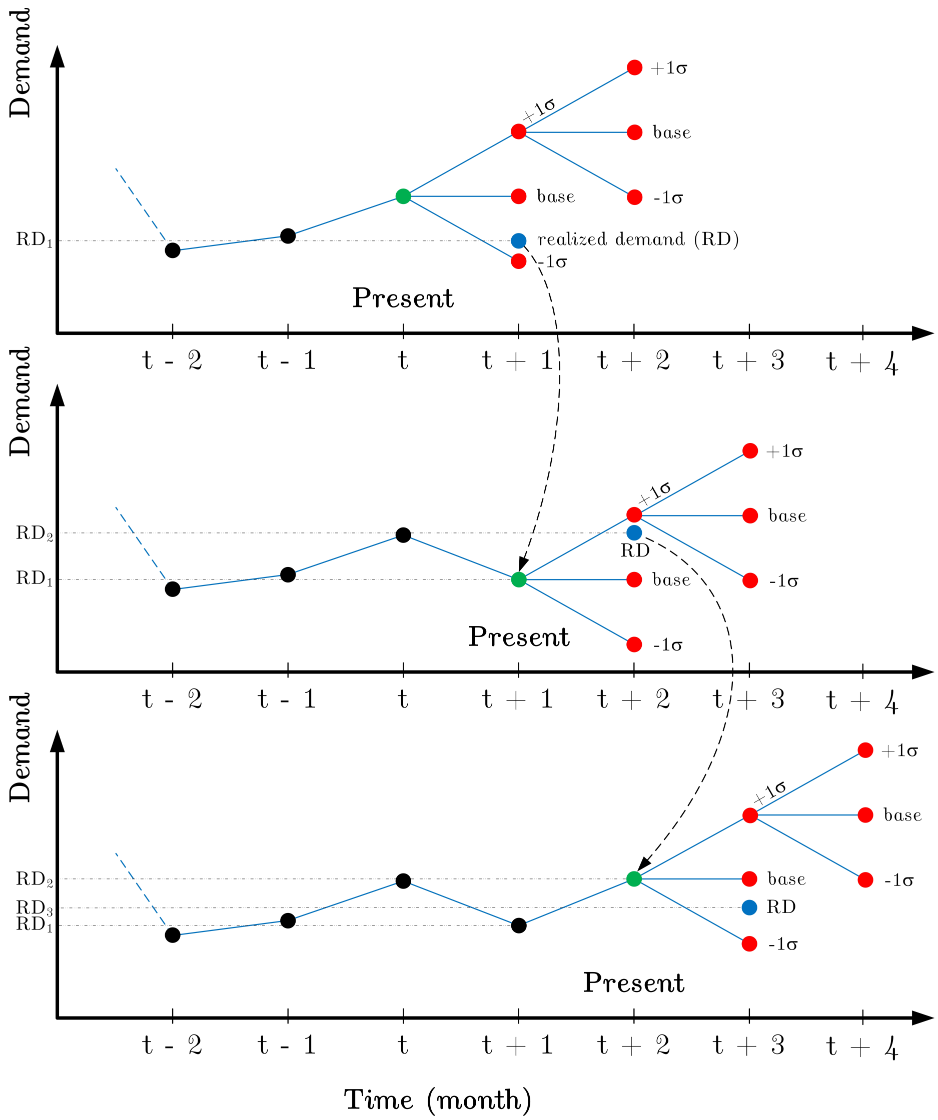

Figure 3: Rolling Horizon Multistage Stochastic Approach

Instruction for Model Validation

We develop our codes in Python 3.6. LP models are solved in a Gurobi Python interface. In order to run the model the following modules should be installed: *pandas, numpy, datetime, statsmodels, gurobipy, scipy.* A summary of all files in the repository are given in the *Read Me* file. Three main files int he repository are "Main.py", "Next-24-Months.csv" and "Results.xls". *Main.py* is the main code to run the program. The default scenario tree is [1, 7, 3, 1] but it can easily be change if needed. Follow this instruction to validate the model with real data:

1. Open **"Next-24-Month.csv"**. Copy and paste the *values* of the next 24 demand points under "x" column. Close the file.
2. Open **Main.py** and hit the run bottom. Wait until complete. It should not take more than one hour in an ordinary lap top.
3. Open **"Results.xls"** and find the required outputs.

References

[1] Bruno A Calfa, Anshul Agarwal, Ignacio E Grossmann, and John M Wassick. Data-driven multi-stage scenario tree generation via statistical property and distribution matching. *Computers & Chemical Engineering*, 68:7–23, 2014.

[2] Yongpei Guan, Shabbir Ahmed, George L Nemhauser, and Andrew J Miller. A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. *Mathematical Programming*, 105(1):55–84, 2006.

[3] Kjetil Høyland, Michal Kaut, and Stein W Wallace. A heuristic for moment-matching scenario generation. *Computational optimization and applications*, 24(2-3):169–185, 2003.

[4] Warren B Powell. What you should know about approximate dynamic programming. *Naval Research Logistics (NRL)*, 56(3):239–249, 2009.

[5] Funda Sahin, Arunachalam Narayanan, and E Powell Robinson. Rolling horizon planning in supply chains: review, implications and directions for future research. *International Journal of Production Research*, 51(18):5413–5436, 2013.

[6] Suresh Sethi and Gerhard Sorger. A theory of rolling horizon decision making. *Annals of Operations Research*, 29(1):387–415, 1991.

[7] Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, 17, 2007.