Paper 2794-2018

Optimizing Inventory of Slow Moving Products Using SAS® Optimization

Lokendra Kumar Devangan, Richard Dansoh and Malleswara Sastry Kanduri, CoreCompete, Amy McArthur, Advance Auto Parts

ABSTRACT

Aiming to use data and science to set service-level goals, Advance Auto Parts engaged CoreCompete to deliver a fully integrated service-level (Inventory) optimization system using SAS® Inventory Optimization and SAS/OR® software. The system has the ability to run inventory simulations and execute large-scale optimization for service-level goal optimization, leveraging the batch services on Amazon Web Services. A very large mixed integer optimization problem for inventory cost reduction is solved using the OPTMODEL procedure. The solution has the ability to recommend optimized service-level goals at the SKU/location level. System design integrates a simple Microsoft Excel user interface, data processing in Apache Hadoop, and optimization in SAS® in the cloud and the dashboards in SAS® Visual Analytics in order to review results. The end-to-end process flow for implementing simulations and optimization in large scale is discussed in this paper.

INTRODUCTION

This paper describes the process to design inventory simulation and optimization system at scale and its integration with existing replenishment system JDA E3. SAS® procedures PROC OPTMODEL and PROC MIRP have been used extensively to optimize inventory and service level goals at SKU/location level. The business problem, solution design, and results will be discussed. Various assumptions made to model the supply chain network of Advance Auto Parts are stated. Implementation challenges including scalability will be discussed. Business constraints have been honored in inventory simulation and optimization. Initial results have shown significant improvement in inventory cost and in-stock rates.

INVENTORY OPTIMIZATION PROBLEM

Advance Auto Parts (AAP) is the second largest auto-parts retailer in the U.S. with over 5,200 stores and annual revenue of $10 billion in 2016. They are presently using E3, a proprietary product of JDA software for managing inventory replenishment at its distribution center (DC) and stores. Service level goals assigned to each item are based on rule of thumb and are not optimal. Auto parts industry in the U.S. serves a massive number of parts for use by a vehicle from a huge number of manufacturers, model and makes, coexisting at the same time. Thus, SKU/Store demand at AAP is severely low and intermittent, where significant portion of SKU/Store combinations observe less than 2 sales per year. Minimizing lost sales as well as inventory holding cost is a challenging problem.

Another challenge faced by AAP is due to magnitude of SKU/location count. With around 80 million SKU/Locations, the scale of the problem makes optimization difficult.

SOLUTION DESIGN

Service level optimization solution has been designed by integrating Elastic MapReduce (EMR) services on Amazon Web Services (AWS) for data processing to SAS® engine for simulation and optimization on AWS batch. Users can interact with the system using Microsoft Excel based interface and analyze the results on SAS® Visual Analytics. Spark on EMR is used to expedite the data processing.

To optimize the service level goal at distribution center (DC) and stores, two independent simulations for each SKU/Store and SKU/DC are designed. The following assumptions are made to model supply chain network for Proc MIRP simulation.

1. DC inventory simulation and store inventory simulations are independent. SKU/Store simulation is also independent of other stores.
2. SKU/Stores and SKU/DC forecast is used to model SKU demand in the MIRP simulation. Variance is same as daily mean demand as Poisson distribution is assumed.
3. Simulation horizon 565 days and 200 days are used as warm up period.
4. Margin is difference between retail price and current unit cost.

Figure 1 below explains the process flow of the system designed.

.

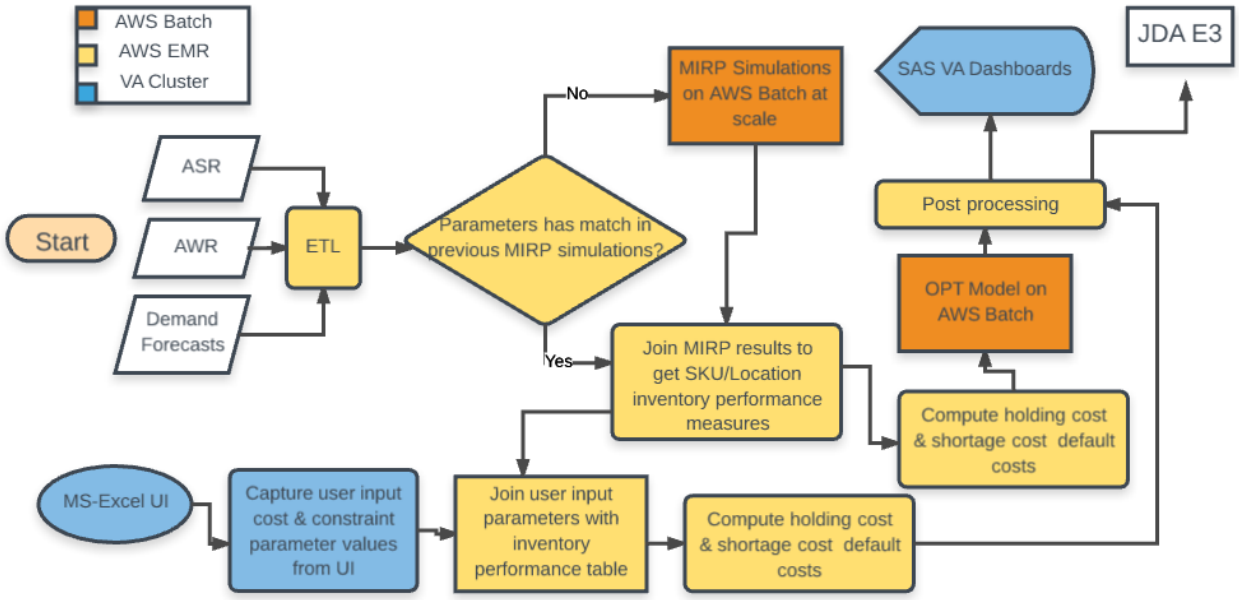


Figure 1 End to end process flow

INVENTORY SIMULATION: SKU/LOCATION

Inventory simulation is designed to observe how will the SKU/location behave for different values of service level goals considering supply chain constraints like SKU demand and variance, lead time etc. SAS® procedure MIRP (Multi Echelon Inventory Replenishment Plan) has been used for designing inventory simulation model. PROC MIRP can be executed with different modeling objective values tuned to different business objective. OPTPOLICY and PREDICTKPI have been used for Advance Auto Parts. OPTPOLICY is used to generate the optimal inventory policies and PREDICTKPI is used to compute the inventory performance metrics such average inventory, fill rate etc. Any supply chain network can be organized in the structure required in PROC MIRP and can be used to compute the inventory control parameters and performance metrics. PROC MIRP procedures requires four datasets as input in a specified structure. Descriptions of three datasets required are provided below in Table 1

Datasets	Description
DemandData	Takes demand and variance at customer facing SKU/location
NodeData	Takes supply chain network parameters such as lead time, PBR, batch size etc. at SKU/location.
ArcData	Takes predecessor and successor information for supply chain network locations
InventoryData	This is not required when objective is OPTPOLICY. It takes inventory policy control values which are output from OPTPOLOCY and initial value of inventory.

Table 1 Input datasets to PROC MIRP

Data for simulation have been prepared using E3 systems, Advance Store Replenishment (ASR) and Advance Warehouse Replenishment (AWR) for inventory replenishment at stores and DC. To simulate the AAP supply chain network, parameters values are taken from ASR, and AWR tables which are listed in the Table 2 below

	Simulation Parameter	Source for store simulation	Source for DC simulation
1	Mean Demand	ASR	AWR
2	Variance	ASR	DC order variance
3	Lead time	ASR	AWR
4	Periods between review(PBR)	ASR	AWR
5	Batch size	ASR	AWR
6	Ordermin	ASR	AWR

Table 2 Simulation data sources

Below is snippet of code, used for executing MIRP procedure for SKU/location for different level of service level goal. This structure is modified for running simulations for thousands of SKU/Stores and SKU/DC on the AWS batch. Each SKU/location is simulated for 565 periods (days). Number of replication used is 2000.

```
Proc mirp nodedata=work.nodedata
     arcdata=work.arcdata
     demanddata=work.demanddata
     out=temp_optpolicy_output
     policyparam=integer
     demandmodel=discrete
     replications=2000
     cv2=yes
     forecastinterval=1
     horizon=565
     maxcv=1
     objective=optpolicy;
run;

proc sql;
  create table optpolicy_output as
   select a.networkid,
          a.skuloc,
```

```
          a.period,
          a.*,
          case when period=1 then initial_inventory else . end as amount
    from temp_optpolicy_output a

      order by networked,skuloc,period;
quit;

Proc mirp nodedata=work.nodedata
     arcdata=work.arcdata
     demanddata=work.demanddata
     inventorydata=optpolicy_output
     out=predictkpi_output
     policyparam=integer
     demandmodel=discrete
     replications=2000
     cv2=yes
     forecastinterval=1
     horizon=565
     maxcv=1
     objective=predictkpi;
run;
```

Independent simulations for SKU/store and SKU/DC at different service levels (81% to 99%) for 565 periods (day) were executed and developed a library of inventory performance metrics for each unique combination of supply chain network parameters. Inventory performance metrics are listed below

1. On hand Mean
2. Backlog Mean
3. Order up to level
4. Reorder level
5. Safety stock
6. Fill rate
7. Ready rate

Output of PROC MIRP with objective = PREDICTKPI produces above measures at period level. We drop measures for the first 200 periods to account for impact of initial inventory in the system and rest are used for computing averages at SKU/Location level. We used a conservative warm-up period since the demand is low; in 200 days, many SKUs would see a demand of only 1 unit.

SERVICE LEVEL OPTIMIZATION: SKU/LOCATION

Optimization model picks the service level which minimizes the sum of holding cost and shortage cost. Parameters used in the optimization are defined below

Holding cost: Holding cost is set by the user and can vary by department, item class and vary between DC and store.

Shortage cost = (1-Fill Rate) x annual demand x product margin, where margin is mode of retail price minus current unit cost.

Indices

i: index for item

j: index for store
k: index for service level at store
q: index for DC
r: index for service level at DC

Decision Variables

x: binary for item-store-SL allocation

y: binary for item-DC-SL allocation

Constants

Total_Cost_Store : Holding cost +Shortage cost at store

Inv_store: Average inventory at store

Inv_cost_store: Inventory cost at store

Total_Cost_DC : Holding cost +Shortage Cost at DC

Inv_DC: Average inventory at DC

Inv_cost_DC: Inventory cost at DC.

Objective:

Minimize: Total Cost (Holding Cost + Shortage Cost) across all SKU-Stores and SKU-DCs

$$\sum_i \sum_i \sum_k (x_{ijk} * \text{Total\_Cost\_Store}_{ijk}) + \sum_i \sum_q \sum_r (y_{iqr} * \text{Total\_Cost\_DC}_{iqr})$$

Decision:

For each SKU-Store and SKU-DC, pick a ready rate. This is binary as only one Ready Rate can be picked. Thus, the number of binary variables will be equal to the sum of the number of ready-rate records across all SKU-Stores and SKU-DCs. We have defined two categories of binary decision variable 1. Store-SKU binary and DC-SKU binary. Hence decision variables

x_{ijk} and y_{iqr} are binary variables

Constraints:

Set 1: Store-SKU and DC-SKU can have only one Service Level. Pick only one ready rate. (ensures only one ready rate is selected for a SKU-Store or SKU-DC

For each SKU-store, sum of binaries = 1

$$\sum_k (x_{ijk}) = 1 \quad for\ i, j$$

For each SKU-DC, sum of binaries = 1

$$\sum_r (y_{iqr}) = 1 \quad for\ i, q$$

When there is no budget constraint model recommends optimum service level without taking account of allocated budget. Note that since we are balancing holding cost and Shortage Cost, optimization will not pick the ready rate of 99% unless justified; the max ready rate might be incurring very high holding cost and not giving us enough reduction in Shortage Cost.

Set 2: Budget for inventory is limited

For all SKU-Locations and their ready rates, total cost of inventory cannot exceed budget

Inventory cost for SKU/Locations will be average inventory at given Ready Rate X Unit cost of the SKU

$$\sum_i \sum_i \sum_k x_{ijk} * \text{Inv\_store}_{ijk} * * \text{Inv\_cost\_store}_{ijk} + \sum_i \sum_i \sum_k x_{ijk} * \text{Inv\_DC}_{ijk} * * \text{Inv\_cost\_DC}_{ijk} \leq Budget$$

PROC OPTMODEL is used to model the above optimization problem which is executed on AWS cloud using batch services.

TECHNICAL CHALLENGES AND RESOLUTIONS

Advance Auto Parts manages tens of millions, SKU/locations. Number of unique SKU covered in this project exceeded 100,000 with thousands of stores. Running PROC MIRP simulation honoring business constraints such as batch size, minimum order and min max inventory policy is highly time-consuming process. MIRP simulation of an independent SKU/location for 15 distinct service level values takes on an average 7 seconds for 2,000 replications and 565 periods on r4.4xlarge having 16 vCPU and 122GB RAM. If number of replication is reduced to 500 it takes 3.5 seconds but at the cost of accuracy. By going with this estimate, it will take years to run independent simulations for millions of SKU/locations.

SCALING OUT SIMULATION AND OPTIMIZATION ON AWS

Solution library and elastic processing on AWS are used to resolve run time challenges. To deal with time consuming exercise of simulation and optimization, PROC MIRP and PROC OPTMODEL are deployed and executed as AWS Batch jobs. Apache Hadoop was used for reducing the taken in data preparation and output summarization. The end to end approach used for simulation and optimization is discussed below.

Run simulation for unique set of supply chain parameters. If 10 SKU/stores have same parameters (lead time, PBR, Mean demand, Variance etc.), run simulations only once for one SKU/store and store inventory performance metrics in a permanent solution library. This reduces number of simulations required significantly close to 2,000,000 from 80 million but still simulation cannot be completed within hours.

Deploy PROC MIRP on AWS Batch services to run simulations in parallel in thousands of machines at cheap cost for SKU/location not having match in the solution library. Candidate SKU/locations with unique parameters can be divided into multiple batch job tables. AWS Batch services forms a queue of jobs and allocates a machine to each job. Machines use preconfigured docker image registered in AWS repository to create required SAS® environment. Data and SAS® program required for the simulation is exported using a shell script.

Inventory Optimization is scaled out using EMR for initial data processing and post optimization data processing for each department. PROC OPTMODEL is deployed on AWS batch services to run N optimization for N departments. The

Figure 2 below depicts architecture for scaling out simulation and optimization

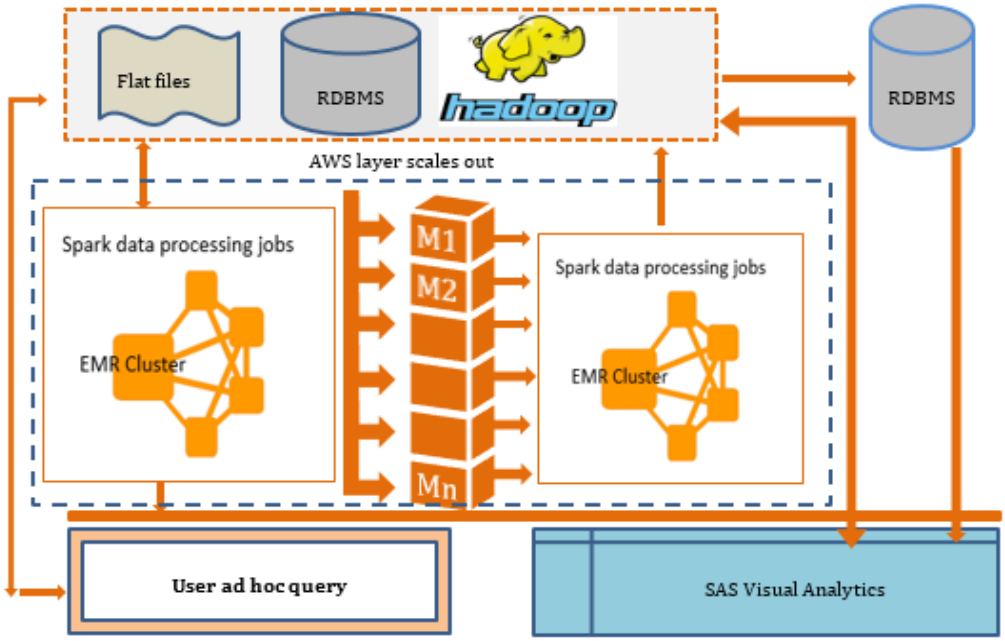


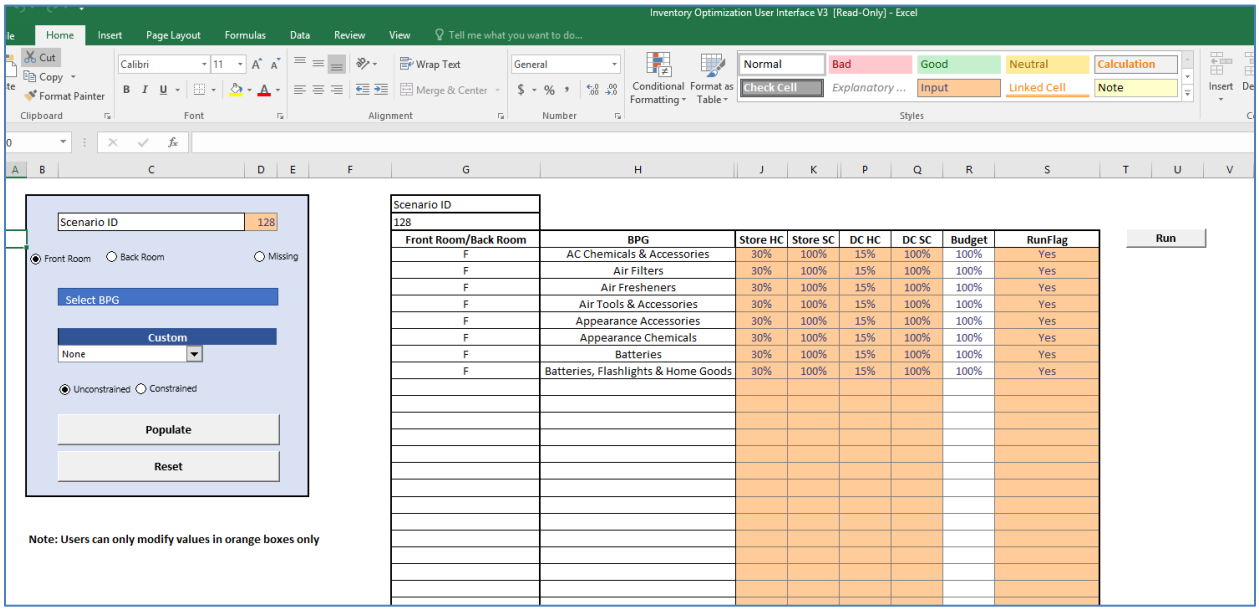
Figure 2 architecture to scale out simulation/optimization

BUSINESS CHALLENGES AND RESOLUTIONS

Business users wants to test multiple scenarios with different assumptions on cost and lost sales. An On-Demand Inventory Optimization tool is developed which can be used to launch optimization for selected departments. Users can run test scenarios using this tool. Microsoft Excel is used for developing user interface and optimization results are shown on SAS® Visual Analytics (VA) platform.

ON DEMAND OPTIMIZATION TOOL: USER INTERFACE

Microsoft Excel based custom user interface enables user to modify carrying cost and lost sales assumptions and analyze the scenario results. Business users can use this custom interface for running on demand optimization for multiple departments on AWS. Users can kick-off optimization with different values of cost and constraint parameters. VBA program is used to capture the input information which is passed into SAS® server on Unix platform using Putty. VBA program triggers shell script which eventually triggers SAS® program. AWS EMRs have been used for ETL process and are smoothly integrated with SAS® engine. SAS® Visual Analytics is used to display the results. Display 1 shows the interface developed using MS Excel.



Front Room/Back Room	BPG	Store HC	Store SC	DC HC	DC SC	Budget	RunFlag
F	AC Chemicals & Accessories	30%	100%	15%	100%	100%	Yes
F	Air Filters	30%	100%	15%	100%	100%	Yes
F	Air Fresheners	30%	100%	15%	100%	100%	Yes
F	Air Tools & Accessories	30%	100%	15%	100%	100%	Yes
F	Appearance Accessories	30%	100%	15%	100%	100%	Yes
F	Appearance Chemicals	30%	100%	15%	100%	100%	Yes
F	Batteries	30%	100%	15%	100%	100%	Yes
F	Batteries, Flashlights & Home Goods	30%	100%	15%	100%	100%	Yes

Display 1 User interface to launch test optimization scenarios

DASHBOARD ON SAS VISUAL ANALYTICS

SAS® Visual Analytics (VA) is used to display the results obtained by the optimization. Reports have capability to compare the multiple scenarios. There are sections to report monetary impacts on stores and DC separately. The cost parameters entered by the user in On Demand Microsoft Excel tool is also made available in SAS® VA.

CONCLUSION

By storing MIRP results, significant computation costs and time can be reduced by creating a library of expected supply chain results at various service levels. Using a containerized approach to parallelizing SAS® in docker images allows Advance Auto Parts to utilize the advantage of elastic computing offered by the cloud to run inventory optimization on a very large supply chain.

The optimization capability allows Advance Auto Parts to allocate constrained budget resources optimally based on the margin, demand velocity and supply chain structure of every store location to maximize expected return from limited inventory investment dollars.
The service level recommendations are highly sensitive to shortage and holding cost. The interface developed allows planners to apply judgement and analytical insights from other teams to adjust the holding and shortage costs used in the optimization. This allows the system to reflect substitutability in demand through lower shortage costs, as well as the very different holding costs associated with store vs. DC, high cube items vs smaller items and other scenarios.
PROC MIRP can be used for developing simulation model for a complex supply chain network. For large network PROC MIRP takes longer time compared to smaller networks. Also, number of replication plays significant role in accuracy as well as in time complexities. These challenges can be handled easily by moving the simulation and optimization on AWS batch which scales out at low cost.

REFERENCES

SAS Institute Inc. 2014. SAS® Inventory Replenishment Planning 2.3: User's Guide. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2011. SAS/OR® 9.3 User's Guide: Mathematical Programming. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

Special thanks to Dr. Ajay Mishra, CoreCompete and Dr. Christopher Houck, CoreCompete for the guidance during the design and development of scalable inventory optimization system. Also, thanks to Jason Blane, Advance Auto Parts and Mark O'Connell, Advance Auto Parts for providing business insights and understanding to develop the solution.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Lokendra Kumar Devangan
CoreCompete
E-mail: Lokendra.devangan@corecompete.com
www.corecompete.com