Proceedings of the 2007 Industrial Engineering Research Conference
G. Bayraksan, W. Lin, Y. Son, and R. Wysk, eds.

Evaluating Clustering Methods for Multi-Echelon (r,Q) Policy Setting

Vikram L. Desai, M.S.; Manuel D. Rossetti, Ph.D., P.E.
Department of Industrial Engineering
University of Arkansas
4207 Bell Engineering Center
Fayetteville, AR 72701

Abstract

In this research, we develop a segmentation methodology for reducing the computational time required for setting the (r, Q) inventory control polices in a large-scale multi echelon inventory system. The segmentation methodology uses a clustering algorithm to compress the inventory data into clusters of identical SKUs at each inventory control point and uses the reduced dataset of clusters for optimizing the control policies. While, the segmentation methodology reduces the computational time required for setting the inventory control policies in a multi echelon inventory system, it also induces a penalty cost in the form of increased inventory investment due to loss of identity for SKUs while clustering. We tested the segmentation methodology on different multi echelon inventory system scenarios to understand the degree to which the computational time can be reduced and to understand the effect of the resulting penalty cost.

Keywords
Inventory policy setting, multi-echelon, clustering

1. Introduction and Background

A single inventory control point in a multi echelon inventory system, stocks a wide range of stock keeping units (SKUs) for a varied customer base. The demand rate across the SKUs may vary from as little as 1unit per year to thousands of units per year. Whereas, the lead times for procuring these SKUs could vary from few days to months. Setting inventory stocking policies for such a diverse set of SKUs held at different inventory control points in order to provide the desired service level at the lowest possible cost is a key aim for many organizations. The stochastic demand, stochastic failures, stochastic lead-times and repair times of the SKUs makes modeling a multi echelon inventory system for setting inventory stocking policies extremely difficult, especially when we are dealing with a large-scale inventory system have hundreds of thousands of SKUs.

Organizations having multi echelon supply chains viz; tech companies like IBM, Hewlett-Packard; retailers such as Wal-Mart, Target, Albertsons; government agencies like the US Navy, US Air-Force etc set stocking policies for their millions of SKUs held at different echelons using commercially available or in-house built inventory optimization software solutions. Due to the complex nature of the inventory control policy setting algorithms in this software, the total computational time required to set policies increases as the number of SKUs increases. For a typical multi echelon inventory system consisting of multiple depots and multiple bases with thousands of SKUs stocked at each location, like the one maintained by the US Navy, the total computational time for setting the inventory control polices using multi echelon inventory optimization software is in hours. Considering the number of businesses around the globe maintaining multi echelon inventory systems and using multi echelon inventory optimization software solutions, there is a serious need to bring down the computational time from hours to minutes. This research examines the use of clustering techniques in combination with policy setting algorithms to reduce the computational time for multi echelon inventory optimization software solutions.

The objective of multi-echelon inventory optimization is often defined in terms of minimizing the system inventory level subject to achieving a desired customer service level. Multi- echelon inventory optimization models examine the entire system, searching for better solutions for the entire chain, not each stage independently. This coordination has the advantage of achieving a better global solution. A full review of multi-echelon policy setting algorithms is beyond the scope of this paper. We refer the interested reader to Al-Rifai´ and Rossetti (2007) and the references therein for an up to date literature summary. Work on combining clustering and policy setting is limited. The primary paper in this area is the work presented by Ernst and Cohen (1990). While developing a model for setting optimal inventory stocking policies for a major automobile manufacturer that stocked over 300,000 part-types in an extensive network with approximately 50 distribution centers and thousands of dealer locations, Ernst and Cohen (1990) developed a segmentation methodology called ORG (Operations Related Groups) as they realized the advantage of grouping the part-types based on operationally relevant attributes and defining generic, group based policies for controlling inventory could be substantial. Rossetti and Achlerkar (2004) illustrated the application of clustering techniques at a single large-scale inventory location.

In what follows, we describe our basic methodology including a brief overview of our own policy setting algorithm. Then, we present a set of experiments to illustrate the effect of clustering on the policy setting procedures. Finally, we describe on-going and future work in this area.

2. Methodology

Without loss of generality, we consider a multi-echelon inventory system as one consisting of two echelons. The lowest echelon consists of bases that experience direct demand from the customers whereas the highest echelon consists of depots which fill the orders placed by the bases in the lowest echelon. We assume that a depot can support multiple bases but each base is supported by one and only one depot and that there is no lateral transshipment between bases or depots. The bases are non-identical, i.e., a part-type stocked in different bases, say Base A and Base B can have different attributes viz., different annual demand, lead-times, etc. A continuous review (r, Q) control policy is assumed at both the echelons.

Our solution methodology uses a clustering algorithm, for reducing the computational time required for setting inventory control policies in multi echelon inventory systems. Clustering is a data-mining tool, used in various applications to identify patterns, compress the data or segment the data. The clustering algorithm employed in the solution methodology, groups the SKUs with similar attributes (Annual demand, Unit Cost, Lead Time, etc) as a single pseudo-item. This reduces the inventory data of hundreds of thousands of SKUs into few thousands of pseudo-items for use in multi echelon inventory optimization software, thus reducing the computational time. With this solution methodology, instead of setting stocking policies for each SKU at all the locations individually, the policy makers would have to deal with a reduced number of pseudo-items. The segmented inventory data generated by this solution methodology would also empower the decision makers with more control over all the SKUs for strategic analysis as the number of SKUs would be reduced from hundreds of thousands to very few pseudo-items. Although clustering of inventory has several advantages from a managerial perspective, there is a trade-off involved. A penalty cost is expected for grouping the SKUs into pseudo-items due to loss of identity as the generic group-based policy (reorder point, reorder quantity) is applied to the SKU instead of the SKU's optimal control policy. This penalty cost is reflected in the total inventory investment of the system. Also, some discrepancy in the service levels of the SKUs is expected due to loss of identity.

In this research, we experiment to see the effects of clustering the inventory data on the entire supply chain's performance metrics. Our aim is to understand the amount of reduction in computational time that could be attained through the use of clustering algorithm and its effects on the total inventory cost, average system fill rate, total system backorders and average system order frequency. The results obtained from the experiments will help in identifying the factors to be used for clustering the inventory dataset. This may allow a reduction in the computational time at the same time maintain the penalty cost within acceptable limits. Out main contribution is our overall methodology which is a combination of well established approaches. Due to space limitations, we will only briefly describe our multi-echelon inventory model, its interaction with our segmentation methodology, and some initial experimental results.

2.1 Multi-Echelon Inventory Model

To set stocking policies for SKUs before clustering and for pseudo-items after clustering, during our experimental procedure, we have developed a multi echelon inventory optimization model. The model we have formulated is based on Duermeyer and Schwarz's (1981) model to evaluate performance measures of a multi- echelon inventory system. They assume a generic *(r, Q)* policy at both retailers and warehouses, as is the case in most of the real world inventory systems and as opposed to *(S-1, S)* base-stock policies pervasive in the literature of multi-echelon inventory systems, which is a special model for low demand and high cost parts. Our model assumes a continuous review *(r, Q)* policy at both the depots and the bases. Our argument is that, not all SKUs stocked in a multi echelon inventory system have high cost, low demand characteristics to apply base-stock policy across all the SKUs. A base-stock policy is the special case of *(r, Q)* policy where reorder quantity, $Q = 1$ and reorder point(r) is base stock level$(S) -1$. Hence, if a SKU has a Low Demand, High Cost characteristic the generic *(r, Q)* policy will automatically assign it a base sock policy with $Q = 1$. The generic *(r, Q)*, multi-item, multi echelon inventory optimization model is given as follows:

- i Set of bases at the lowest echelon, $\{1\ldots M\}$
- k Set of depots at the highest echelon, $\{1\ldots L\}$
- j Set of part-types stocked in the system, $\{1\ldots N\}$
- C_j Unit cost of part-type j
- $\lambda_{i,j}$ Demand Rate of part-type j at base i
- $\lambda_{k,j}$ Demand Rate of part-type j at depot k
- F_k Target average order frequency at depot k
- F_i Target average order frequency at base i
- B_i Target backorder level at base i
- B_k Target backorder level at depot k
- $I(r_{k,j}, Q_{k,j})$ Avg. inventory of part-type j at depot k
- $I(r_{i,j}, Q_{i,j})$ Avg. inventory of part-type j at base i
- $B(r_{k,j}, Q_{k,j})$ Avg. backorder level of part-type j at depot k
- $B(r_{i,j}, Q_{i,j})$ Avg backorder level of part-type j at base i

$$\text{Min } \sum_{k=1}^{L}\sum_{j=1}^{N} C_j \times I(r_{k,j}, Q_{k,j}) + \sum_{i=1}^{M}\sum_{j=1}^{N} C_j \times I(r_{i,j}, Q_{i,j})$$

$$\text{s.t. } \frac{1}{N}\sum_{j=1}^{N}\frac{\lambda_{k,j}}{Q_{k,j}} \le F_k \quad \forall \quad k = 1...L$$

$$\frac{1}{N}\sum_{j=1}^{N}\frac{\lambda_{i,j}}{Q_{i,j}} \le F_i \quad \forall \quad i = 1...M$$

$$\sum_{j=1}^{N} B(r_{k,j}, Q_{k,j}) \le B_k \quad \forall \quad k = 1..L$$

$$\sum_{j=1}^{N} B(r_{i,j}, Q_{i,j}) \le B_i \quad \forall \quad i = 1..M$$

$$r_{i,j}, r_{k,j} \ge 1 \qquad Q_{i,j}, Q_{k,j} > 0$$

- $r_{k,j}$ Reorder point for part-type j at depot k
- $Q_{k,j}$ Reorder quantity for part-type j at depot k
- $r_{i,j}$ Reorder point for part-type j at base i
- $Q_{i,j}$ Reorder quantity for part-type j at base i

In order to be more consistent with real world multi-echelon inventory systems, we allow non-identical bases at the lowest echelon, meaning each base at the lowest echelon stocks varied number of part-types and common part-types have different characteristics and attributes at different base locations. The customer-demand for part-type j at base i is stochastic and assumed to be Poisson with mean $\lambda_{i,j}$. As the inventory position reaches $r_{i,j}$ an order of size $Q_{i,j}$ is placed at the respective depot k. The base lead times are assumed to be the sum of two components: fixed transportation time and average backorder waiting time. Any order that is not met at the base is backordered and is fulfilled on first come first serve (FCFS) basis. We approximate the average backorder waiting time using the METRIC (Sherbrooke, 1986) approximation as is commonly done in the literature. At the depot, if the on-hand inventory is greater than or equal to the demanded order quantity $Q_{i,j}$, the order is fulfilled or else the entire order is backordered as no order-splitting is allowed. As the inventory position at the depot reaches $r_{k,j}$ an order of size $Q_{k,j}$ is placed at the depot's supplier which is received after a constant transportation time i.e., depot lead time, as we assume that the supplier is with infinite resource and replenishes the depot immediately. Constraints are included on the average order frequency and the total backorder level for depots and bases respectively. The *(r, Q)* policy setting model is an integral part of our experimental procedure and serves as a tool for comparative analysis to test the affects of clustering the SKUs on the performance measures of the system. We do not aim at developing an optimization model that would set global optimal stocking policies for SKUs in a multi echelon inventory system. For our purposes, an inventory control policy setting model generating near-optimal solution is sufficient. From this research's point of view, the *(r, Q)* policy setting model is solely used for comparative analysis of the performance measures before and after clustering. The following section, explains our methodology to set stocking policies at each location in the echelon.

2.2 Segmentation Methodology

The step-by-step procedure for setting stocking policies in the entire network using clustering concept for reducing the computational time is as follows:

1) For each base {1, 2…, M} in a network, compute the initial values for order quantities $Q_{i,j}$ for each part-type $j \in \{1, \ldots N\}$ based on the average order frequency constraint F_i at that base, $Q_{i,j} = \lambda_{i,j} / F_i \ \forall \ j = 1 \ to \ N \ and \ \forall \ i = 1 \ to \ M$
2) Compute the expected lead-time demand (in terms of orders) and variance of the lead time demand for each part-type j at depot k of the network. See Desai (2006) for the detailed formulas.
3) Use SAS's *FastClus* procedure (*K Mean* clustering algorithm) to compress the depot's part-type dataset into the desired number of cluster (pseudo items) in terms of percentage of actual dataset. The following attributes of the part-types stocked at the depot are used for clustering: unit cost, demand rate, depot lead time, base location index.
4) Set the attributes for pseudo items (clusters) based on the Group Policy Deciding Criteria (Min, Avg, Max). If the Group policy deciding criteria at a particular design point of the experimental setup is "Average" then the attributes of the cluster (pseudo item) will have the average attribute value of all the part types in the cluster.
5) Set the stocking policies $(r_{k,j}, Q_{k,j})$ for all the pseudo items (clusters) at the depot, $K \in \{1,2,\ldots,L\}$ of the network, using Multi-Product Backorder Model (Hopp, Spearman and Zhang, 1995) so as to minimize the total inventory cost subject to the depot service constraints.
6) Apply the generic group based inventory control policies of the pseudo items (clusters) to all the part-types inside it. Expand the clustered dataset back to the original dataset.
7) Compute the average backorder waiting time, BWT_j, for all the part-types j, stocked in depot k of the network under consideration. See formulas in Desai (2006)
8) The $(r_{k,j}, Q_{k,j})$ policies set for pseudo items (clusters) at depot k are in terms of orders and not in terms of unit item. In order to set policies in terms of unit item, we multiply both reorder point $(r_{k,j})$ and reorder quantity $(Q_{k,j})$ with the smallest of the order quantity, $Min(Q_i)_j$ received at the depot from all the bases for part-type j
9) Compute the effective base lead time, L_{ij}, for an order placed by base i for part-type j, by including the expected backorder waiting time, BWT_j for part-type j, at depot K
10) Use SAS's *FastClus* procedure (*K Mean* clustering algorithm) to compress the base's SKU dataset into the desired number of cluster in terms of percentage of actual dataset.
11) Set the attributes for Pseudo items (clusters) based on the Group Policy Deciding Criteria (Min, Avg, Max). If the Group policy deciding criteria at a particular design point of the experimental setup is "Average" then the attributes of the cluster (pseudo item) will have the average attribute value of all the SKUs in the cluster.
12) With the updated lead-time L_{ij}, set the stocking policies $(r_{i,j}, Q_{i,j})$ for all the part-types stocked at base i in the network, using Multi-Product Backorder Model (Hopp, Spearman and Zhang 1995) so as to minimize the total inventory cost subject to the base service constraints.
13) Apply the generic group based inventory control policies of the pseudo items (clusters) to all the SKUs inside it. Expand the clustered dataset back to the original dataset.
14) Repeat step 1-13 for all the bases in the network under consideration

With optimized base order quantities $(Q_{i,j})$ in Step 1, Iterate 1-14, by updating effective base lead-time L_{ij} at each iteration until the effective lead time, L_{ij}, for 75% of SKUs in the network converges within an absolute percentage difference of 0.10 from its value of the previous iteration. This methodology allows the user to define the number of clusters he/she would like the dataset to be reduced to, for setting the inventory control policies. The computational time required for setting the inventory control policies by using clustered dataset through this segmentation methodology would be considerably less than the computational time required using the *(r, Q)* policy setting model.

3. Illustrative Experiments

In this section, we briefly discuss the results of applying the methodology overviewed in Section 2 to a set of illustrative cases. We experimented with various multi echelon scenarios varying the number of SKUs stocked at each location and the number of bases supported by a single depot, while observing the response variables "Reduction in Time (% difference)" and "Penalty Cost (% difference)".

In order to identify the levels of the clustering factors that would reduce the computational time for setting the inventory control polices while maintaining the penalty cost within acceptable limits, we set up a full factorial experimental design with two factors, viz; “Group Policy Deciding Criteria” and “Clusters (%)”. The factor “Clusters (%)” indicates the percentage to which the inventory dataset at each inventory control point is reduced to after using the segmentation methodology. Factor “Group Policy Deciding Criteria” determines how the attributes of the pseudo items (Clusters) formed, are assigned the value after the dataset is clustered. Based on the “Group Policy Deciding Criteria” set at each level of the experiment, the attribute of a pseudo-item can take the minimum, maximum or average attribute value of all the SKUs within the cluster.

Figure 1 shows the average penalty cost observed at various levels of factor “Clusters (%)” with the “Group Policy Deciding Criteria” factor set at “Mean”. We observed that clusters (%) set at 60% results lowest penalty cost of 7.50%.

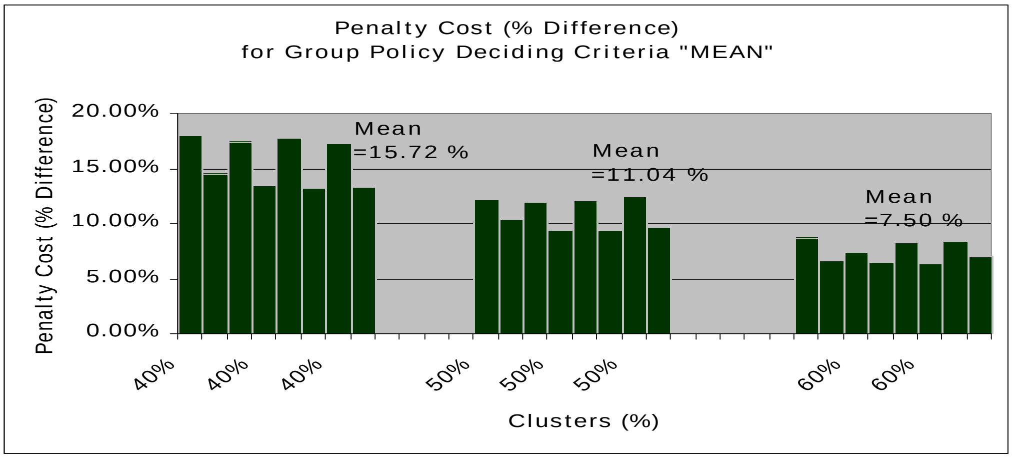

Figure 1: Penalty Cost (% Difference) for Group Policy Deciding Criteria "MEAN"

Our experiments revealed that as the Factor “Cluster (%)” decreases the total computational time required for policy setting also decreases, while maintaining the Order Frequency constraint and Total Backorder constraint at the bases and the depot. On average a reduction of 55.92% in computational time can be achieved when the inventory dataset at each location is reduced to 40% of its original size using the segmentation methodology. Table 1 illustrates ten experiments ran on different multi echelon scenarios with factor “Clusters (%)” set at 40%.

Table 1: Reduction in Time (% Difference) with Clusters (%) = 40%

Scenario	**Cluster (%)**	**Group Policy Deciding Criteria**	**Reduction in Time (% Difference)**
1	40%	Maximum	51.29%
1	40%	Mean	51.69%
2	40%	Maximum	64.37%
2	40%	Mean	64.30%
3	40%	Maximum	48.26%
3	40%	Mean	45.76%
4	40%	Maximum	62.60%
4	40%	Mean	64.12%
5	40%	Maximum	47.30%
5	40%	Mean	47.46%

4. Conclusions and Future Research

With the results observed through experimentation, we can conclude that the segmentation methodology is a potent tool for reducing the computational time required for setting the (r, Q) inventory control polices in large and diverse multi echelon inventory systems. We also observed that the segmentation methodology does not negatively affect the supply chain performance metrics. With the clustering algorithm and the factors explored in this research, we showed that the computational time can be reduced by about 55.92% by using the segmentation methodology. But with this reduction in the computational time we also observed a penalty cost of about 15.72% for using the segmentation methodology. The penalty cost (% Difference) resulting from using the segmentation methodology is not drastically high, but we recommend future work to further bring down the resulting penalty cost.

For future research, we recommend the exploration of additional clustering algorithms so as to use the most suitable algorithm and improve the quality of groups obtained using the segmentation methodology. For experimentation, we explored “Maximum”, “Mean” and “Minimum” as group policy deciding criteria for the clusters formed using the segmentation methodology. Where, “Minimum” failed to maintain the targeted performance constraints for most of the scenarios, we observed the smallest penalty cost for using the segmentation methodology was obtained using “Mean” as the group policy deciding criteria. We recommend experimenting with “Mode” and “Median” as group policy deciding criteria for the clusters formed using the segmentation methodology to see its effects on the penalty cost. The attributes selected to cluster the SKUs at the base; unit cost, demand rate, base lead time and depot lead time are based on the screening experiments performed by Achlerkar (2004), which indicated their significance in clustering at a single location. We recommend examining other attributes such as a base location index (BLI), order frequency constraint index (OFI), and back order constraint index (BCI). The BLI indicates how many locations a part type has in common, the BCI is the minimum of the total back order constraint across the locations for a part type, and the OFI is the minimum order frequency constraint across the locations for a part type. We also recommend investigating other possible attributes for clustering SKUs at bases and depots to improve the quality of clusters formed and further reduce the penalty cost.

The quality of the output given by the multi echelon inventory optimization software solutions that set optimized (r, Q) or (s, S) inventory control policies in a multi-echelon inventory system depends on the initial values used for reorder point (r), reorder quantity (Q) or Order-Up to-Level (S). Instead of assuming initial values for hundreds of thousands of SKUs in the multi echelon inventory system, we recommend using the solution provided by the segmentation methodology for setting the initial values. We observed that the segmentation methodology maintains the system performance constraints while clustering the inventory data for reducing the computational time. Hence, the initial values set using the segmentation methodology will be within the constraints and provide a good starting point to further optimize the inventory control policy parameters.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0214478. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Achlerkar, A. (2004) “Evaluation of Segmentation Techniques for Spare Parts Inventory Management”, unpublished Master’s Thesis, Department of Industrial Engineering, University of Arkansas, Fayetteville, Arkansas.
2. Al-Rifai´ M. H. and Rossetti, M. D. (2007) “An Efficient Heuristic Optimization Algorithm for a Two-Echelon (R, Q) Inventory System”, to appear in the *International Journal of Production Economics*
3. Deuermeyer, B., and L. B. Schwarz (1981), “A Model for the Analysis of System Service Level in Warehouse/ Retailer Distribution Systems: the Identical Retailer Case”, in: L. B. Schwarz (Ed.), *Studies in Management Sciences*, Vol. 16, *Multi-Level Production/ Inventory Control Systems*, North-Holland, Amsterdam, pp. 163-193.
4. Ernst, R., and Cohen, M.A. (1990), “Operations Related Groups (ORGs): A Clustering Procedure for Production / Inventory Systems”, *Journal of Operations Management*, 9(4), 574-598.
5. Hopp, W. J.; Zhang, R. Q.; and Spearman, M. L., 1999. An easily implementable hierarchical heuristic for a two-echelon spare parts distribution system, *IIE Transactions*, 31, 977-988.
6. Rossetti, M. and Achlerkar, A. (2004) “A Constrained Clustering Algorithm for Spare Parts Segmentation,” *The Proceedings of the 2004 Industrial Engineering Research Conference*, R. King, B. Norman (eds.), Houston, TX.
7. Sherbrooke, C.C. (1986), “VARI-METRIC: Improved Approximations for Multi-Indenture, Multi-Echelon Availability Models”, Operations Research 34 (2), 311-340.