AARHUS UNIVERSITY

Coversheet

This is the accepted manuscript (post-print version) of the article.

Contentwise, the accepted manuscript version is identical to the final published version, but there may be differences in typography and layout.

How to cite this publication

Please cite the final published version:

Larsen, C. (Accepted/In press). A heuristic joint replenishment policy for the case of heterogeneity among items. International Journal of Production Economics.

<https://doi.org/10.1016/j.ijpe.2017.10.019>

Publication metadata

Title:	A heuristic joint replenishment policy for the case of heterogeneity among items
Author(s):	Larsen, Christian
Journal:	International Journal of Production Economics
DOI/Link:	https://doi.org/10.1016/j.ijpe.2017.10.019
Document version:	Accepted manuscript (post-print)
Document license:	CC-BY-NC-ND

General Rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

If the document is published under a Creative Commons license, this applies instead of the general rights.

A HEURISTIC JOINT REPLINISHMENT POLICY FOR THE CASE OF HETEROGENEITY AMONG ITEMS

Christian Larsen

CORAL, Department of Economics and Business, Aarhus University,
Fuglesangs Allé 4, DK 8210 Aarhus V, Denmark

Abstract

The $Q(s,S)$ policy is considered among the best for solving the well-known stochastic joint replenishment problem. However, it has some shortcomings if much heterogeneity exists among the items. Consequently, a new heuristic control policy is constructed. The cornerstone is a policy iteration improvement step, assuming a single possibility for making a joint order and thus to deviate from the rule that the items are governed by independent, re-order and order-up-to, (s,S) policies. But this policy iteration improvement step is done repeatedly at each demand epoch. It is only required the development of one-dimensional value functions of state variables in order to accomplish the policy iteration improvement step. Knowledge about good order-up-to values is important. Furthermore, a relaxation parameter α is introduced such that one can, make it easier or more difficult to issue a joint order, when following the policy iteration improvement step. Finally, in case a joint order is made, a vector of can-order points is used to include additional items. The policy is denoted a (c,S,α) policy. Numerical results shows that the new policy performs significantly better than the $Q(s,S)$ policy for cases with heterogeneity among items. Furthermore, from a given optimal can-order policy, one can construct a better (c,S,α) policy.

Keywords

Inventory control, stochastic demand, joint replenishment, heterogeneity.

1. Introduction

The stochastic joint replenishment problem concerns a multi-item inventory system with $N > I$ items facing random demands. Most often, as in this paper, it is assumed that item-demands are independent Poisson processes, and the demand rate for item j is denoted λ_j . Each time a replenishment order is made, an item-specific replenishment cost k_j is incurred for each item j , $j=1,2,..,N$, in the replenishment order. In addition, in case at least one item is included in the order, a joint replenishment cost K is incurred, irrespective of how many items (and how much is ordered) that are included in the replenishment order. It is the presence of this cost parameter K that triggers the need to develop a coordinated inventory control policy. Each item j on the inventory is incurred a holding cost at rate h_j per unit. All unfilled demand is backlogged and each backlogged unit of item j is incurred a penalty cost at rate p_j . Also, at the time when a unit of item j is backlogged, a fixed penalty cost π_j is incurred. If a replenishment order is issued for item j , it has a constant lead-time L_j . This stochastic joint replenishment problem has received considerable attention in the literature where the focus has been on analyzing good control policies. The can-order policy is a prominent example. It was first analyzed by Balinfy (1964) and subsequently by Silver (1981), Federgruen et al. (1984), Schultz and Johansen (1999), Melchioris (2002) and Johansen and Melchioris (2003). The main problem with the can-order policy is to develop a valid mathematical model. The problem is that the process of special order opportunities (used to decompose the problem into single-item problems) does not follow a Poisson process but is much more “ugly”, see Figures 1-2 in Schultz and Johansen (1999). As documentation for that this can cause a severe bias, consult Table 1 in Schultz and Johansen where it follows that for the 12 item problem reported in Federgruen et al. (1984), the simulated costs of their optimal policy is 2364 and not 2620 as they compute the costs to be. Thus any computational model of the can-order policy must be validated by a simulation model as well, and most often some significant discrepancy prevails between the prediction of the computational model and the result of the simulation model. One can also observe this phenomenon by examining Table 2 in Melchioris (2002). Others have proposed policies where the coordination of replenishment decisions is secured by only being allowed to make replenishment decisions at certain time points. This is accomplished by either having deterministic review intervals or a common stochastic review interval governed by the total demand since the last replenishment opportunity. Base-stock policies based on deterministic review intervals are developed by Atkins and Iyogun (1988). If instead of a deterministic review interval one lets it be dependent on the total demand since last replenishment opportunity, then one gets the QS policy proposed by Reenberg and Planche (1967) and subsequently analyzed by Pantumsinchai (1992). Viswanathan (1997) formulated a $P(s,S)$ policy with a common deterministic review interval and the replenishment decision of each item is governed by an (s,S)

policy. He also proposed an alternative to the $P(s,S)$ policy, which he called the $Q(s,S)$ policy, where the review interval is stochastic as described above. For this policy, Nielsen and Larsen (2005) (see also Larsen (2009)), developed a mathematical model and an algorithm to compute the optimal policy variables. So far, the $Q(s,S)$ policy in general seems to perform best with regard to cost minimization, compared to other proposed policies, when using datasets similar to those of Atkins and Ioygun (1988), that is, when there is a high degree of homogeneity among the items. (The reported performance of (Q,S,T) policy in Özkaya et al. (2006) should be disregarded because the results in that paper are erroneous, as briefly reported by Larsen (2011) - for further elaborations on this, see Larsen (2008)) However, it can be questioned whether the $Q(s,S)$ policy make sense if there is much heterogeneity among the items, for a discussion of this see Johansen and Thorstenson (2006). More alarmingly, if one applies the $Q(s,S)$ policy on such datasets, it might yield poor results, see Dance et al. (2012).

In our paper we show how to modify a given $Q(s,S)$ policy (it could also be a can-order policy) to another policy denoted (c,S,α) . The policy is not as straightforward to explain as other policies since the decision about what to do requires some calculations. However, this decision logic is easy to program on a computer with very small execution time. This is why our proposed policy can only be tested in a simulation model working along a sample path. This idea is very much aligned with ideas of ‘approximate dynamic programming’ promoted by Powell (2011). When applying this new (c,S,α) policy on datasets with much heterogeneity among items, including a dataset of Dance et al. (2011), we can demonstrate significant improvement to the performance of the $Q(s,S)$ policy. Furthermore, we show that all the optimal can-order policies presented in Melchioris (2002) can be improved by our new policy.

In Section 2 we review some of the shortcomings of the $Q(s,S)$ policy and some of the findings of Johansen and Thorstenson (2006) and Dance et al. (2012). In Sections 3 and 4 we will in mathematical terms present the idea for our heuristic, which is based on the policy iteration algorithm, Tijms (2003). In Section 5 we outline the development of the simulation model necessary to perform numerical testing. In Section 6 we carry out the numerical testing on our datasets. We first investigate a very well-known dataset which is analyzed in almost all papers on the stochastic joint replenishment problem. Thereafter we focus on datasets, in which some items are expensive but with low demand while others are inexpensive but with high demand. For these datasets we report on the performance of the (c,S,α) policy compared to the optimal $Q(s,S)$ policy. We finalize the section by demonstrating the performance of the (c,S,α) policy on a given dataset of Dance et al. (2012). Then in Section 7 we review the datasets of Melchioris (2002) and show for each of these how to find a (c,S,α) policy that improves the optimal can-order policy. Finally, Section 8 contains some concluding remarks.

2. Observations about the $Q(s,S)$ policy

The $Q(s,S)$ policy studied by Nielsen and Larsen (2005) works as follows. Each time the aggregate demand since the last order instant has reached level Q , then the inventory is reviewed. Any item j with an inventory position less than or equal s_j has its inventory position brought to level S_j . The solution methodology is based on a decomposition principle, where the joint order cost K is assigned to the stochastic review periods and assumed to be incurred at each review instant. For each item j , the optimal (s_j, S_j) is determined by an adaptation of the algorithm of Zheng and Federgruen (1991) using k_j as the order cost and assuming a stochastic review interval (governed by the control variable Q). The optimal $Q(s,S)$ policy can be found by varying Q .

A major critique against the $Q(s,S)$ policy is that it counts all items equally in the specification of the control parameter Q . This can be particularly disturbing if there is much heterogeneity among the items. The 4-item dataset seen in Table 1 is from Dance et al. (2012).

<Table 1 about here>

There is a clear distinction items $\{1,2\}$ and items $\{3,4\}$, so it might be odd to apply the $Q(s,S)$ policy. This policy has optimal solution $Q = 2$, $s = (-1, -1, -1, -1)$, $S = (1, 1, 0, 0)$ with a computed cost rate 56.56. Because it can happen at a review instant, that all inventory positions are above their re-order levels, the joint order cost K will actually not be incurred at every review instant. Therefore, the actual cost rate will be slightly below 56.56. But simulations show that it is still significantly higher than the cost performance of the independent solution, where each item j is governed independently by an (s,S) policy and where an order cost of $k_j + K$ is incurred. This independent policy has a cost rate of 39.94 (Dance et al. (2012) report this to be 38.31). So from a managerial or an economic point of view it makes no sense to use the $Q(s,S)$ policy in this setting. Dance et al. (2012) propose a policy where the expensive items 3 and 4 are operated by independent base stock policies, while items 1 and 2 are operated by a $Q(s,S)$ policy. Dance et al. (2012) denote this policy $Q_I(s,S)$ and report the optimal policy to have a cost rate of 36.35 (unfortunately the policy variables are not disclosed, so it is not possible to confirm the result). Of course one can argue whether for such an extreme dataset one should at all apply a joint replenishment policy as the result of Dance et al. (2012) is not far from what could have been achieved by applying independent inventory control. Another limitation of the proposal by Dance et al. (2012) is that they only develop an algorithm for the case where the expensive items are controlled by independent base stock policies. In their Section 5 they do propose a more general policy, but apparently do not attempt to convert the general idea into an algorithm.

Johansen and Thorstenson (2006) are also intrigued by the fact that the $Q(s,S)$ policy may be unwise to use. They propose a so-called value-based $Q(s,S)$ policy. Assume you are using the $Q(s,S)$ policy, and at a review instant you have an item whose inventory position is just a few units above its re-order level. If Q is rather large, it could take a long time before a new order opportunity occurs, and therefore the inventory position for this item could become very small. It calls for the need to be able to overrule the $Q(s,S)$ policy. This is exactly the idea of Johansen and Thorstenson (2006). The way they do it, is by making a state specification $(r_1, \dots, r_N)$ where r_j is the number of review instances (except the most recent one) since item j was included in the replenishment order. After having specified the relative values of any state $(r_1, \dots, r_N)$ it is possible to compute, based on the policy iteration algorithm, whether it is optimal temporarily to deviate from the $Q(s,S)$ policy. The point is, however, that even if the number of values that each state variable r_j can attain, appear rather modest, the state space of $(r_1, \dots, r_N)$ will grow dramatically. (The upper bound on r_j can be derived by calculating when the probability of the aggregate demand over consecutive stochastic review periods (which is a multinomial distribution) is at or above level $S_j - s_j$ is almost 1). As an example, dataset no. 6 in Table 1 of Melchior (2002) has each r_j to attain the values 0, 1, 2 or 3 making the total number of states 4^{12} which is a very large number. This may be the reason why Johansen and Thorstenson (2006) refrain from any computations on this dataset. Nevertheless, the idea and motivation behind the approach of Johansen and Thorstenson (2006) is very sound. The key characteristic in their approach is that they offer an adaptive decision making approach. While the traditional inventory control model applies a sort of prepared table, containing specifications of what to do in any state encountered, the adaptive approach considers what to do along any actual sample path: whether one should follow the recommendations in the above mentioned table or whether it is optimal to deviate from this. This approach has similarity to ideas presented in Axsäter (2003), Axsäter (2007) and Tijms

(2003; Section 7.5). The idea of Johansen and Thorstenson (2006) is also pursued in this paper, albeit using a more simplified way to compute the relative values necessary for performing the one-step policy optimization. The price paid for this simplification is that we cannot guarantee to improve the $Q(s,S)$ policy, as Johansen and Thorstenson (2006) can.

The purpose with this section is to make the reader aware of some critical issues for developing optimal continuous review policies for the stochastic joint replenishment problem. On the one hand you have the can-order policy which is easy to explain to practitioners, but it is not so easy to develop a good computational model for how to compute its optimal control parameters. On the other hand you have the $Q(s,S)$ policy containing a fast algorithm for doing the computations, but it may be hard to explain the logic of this policy to a practitioner and more problematically, it may give less good results when heterogeneity is seen among the items. It is with the motivation to bridge this gap, that we have developed our new policy.

3. The general idea behind the heuristic

We denote our control policy a (c,S,α) policy. The vectors of can-order points $(c_1, \dots, c_N)$ and the vector of order-up-to values $(S_1, \dots, S_N)$ are predetermined and are the inputs to the policy. They could be found from an optimal $Q(s,S)$ policy where the can-order points of our policy are the re-order points $(s_1, \dots, s_N)$ of the $Q(s,S)$ policy. They could also be given from a can-order policy (in case a good policy is available). For a graphical outline of the algorithm that constitutes the policy, see Figure 1. The first input $(c_1, \dots, c_N)$ enters directly into the “engine” of the algorithm, which is box no. 3 in Figure 1 while the second input $(S_1, \dots, S_N)$ enters into box no. 2 of Figure 1, where one-dimensional value functions, $R_j(\cdot)$, $j=1, \dots, N$, are constructed, which then together with the vector $(S_1, \dots, S_N)$ are inputs to the “engine”. The third input is a positive parameter α which must be determined experimentally by trial and error. This part is not depicted in the figure.

<Figure 1 about here>

Based on the input $(S_1, \dots, S_N)$, an allocated order cost $\bar{K}_j$ is constructed for each item j , that would make S_j the optimal order-up-to value if item j is governed independently from all other items using an (s,S) control policy and having an order cost $\bar{K}_j$. Given S_j , it is possible from the knowledge about how the algorithm for constructing an optimal (s,S) policy works, see Federgruen and Zheng (1992), to compute $\bar{K}_j$. It is also possible to construct the function of relative values, denoted the value function. This is explained in more detail in Section 4. We now go to the “engine” part, box no. 3 of the algorithm. Assume we observe our inventory system in state $(x_1, \dots, x_N)$, that is, the inventory position of item j , $j=1, \dots, N$, is x_j . Due to the design of our control policy, it always holds that for any j , $x_j \leq S_j$. In a similar spirit as in Axsäter (2003), Axsäter (2007) and Tijms (2003; Section 7.5) we make a decision rule that will minimize the expected costs, using the true cost structure, under the assumption that there is only a single possibility for making a joint order (but the rule is used repeatedly as a heuristic). It means solving an optimization problem of the form

$$\text{Min} \left\{ KI_{\{\exists j: a_j > x_j\}} + \sum_{j=1}^N \left[k_j I_{\{a_j > x_j\}} + R_j(a_j) \right] : a_j \geq x_j, j = 1, \dots, N \right\} \quad (1)$$

Here is I_B an indicator function taking the value of 1 if condition B is true and 0 otherwise. The vector $(a_1, \dots, a_N)$ contains the decision variables of type order-up-to, why the constraints $a_j \geq x_j$, $j = 1, \dots, N$

in (1). Apparently, (1) assumes a more general decision policy due to the presence of the general decision variables a_j . However, due to the way the value functions are constructed, each of the variables $a_j, j=1, \dots, N$, can only attain two values x_j (do nothing) or S_j (make a replenishment order that brings the inventory position of item j to level S_j). The functions $R_j(a_j)$ specifies the expected costs (in excess of the minimum costs specified by the stand-alone (s, S) policy) until the stand-alone (s, S) policy again will trigger an order. See again Section 4 for details. It is also shown in Section 4 that the functions $R_j(a_j)$ have the characteristic to be non-negative and that $R_j(S_j) = 0$. Therefore, according to (1) a joint order should be made if

$$K \leq \sum_{j=1}^N [R_j(x_j) - k_j]^+ \quad (2)$$

Here b^+ is the usual notation for $\max(b, 0)$. The right side in (2) is the aggregate cost saving, across all items, for making orders using the real item-specific order costs. This must be higher than the joint order costs in order to make it optimal to issue a joint order. In that case all items with $R_j(x_j) \geq k_j$ are brought to the inventory position S_j and otherwise the optimal actions are to set $a_j = x_j, j=1, \dots, N$. That is, the set of items that must be ordered is

$$\{j = 1, \dots, N: R_j(x_j) \geq k_j\} \quad (3)$$

There is a catch here, though. Of course, this joint order opportunity is not a once in a life time opportunity, but it is supposed to be used repeatedly after each demand epoch. Therefore the value functions $R_j()$, that are the main inputs to (1), are based on flawed assumptions, and only approximations of the true but unknown N -dimensional value function. Accordingly, given the approximate nature behind the construction of the one-dimensional value functions, it is too rigid to apply (2) and (3). Therefore, we adapt the decision rule in two ways. First, the determination about, whether an order should be issued or not, is not based on (2), but on:

$$K \leq \alpha \sum_{j=1}^N [R_j(x_j) - k_j]^+ \quad (2')$$

The parameter α is positive but might deviate from 1, thereby (2') is a generalization of (2). For examples of how much α in reality can be changed from 1, see our numerical results, reported later. Secondly, similar to the concerns of Johansen and Thorstenson (2006), as a long time may go before a new joint order is made, we allow more items to be included in the order basket, in addition to the set of must-order items from (3). The set of additional items to include in the order basket is:

$$\{j = 1, \dots, N: R_j(x_j) < k_j, x_j \leq c_j\} \quad (4)$$

That is, we include the items whose inventory positions are at or below their can-order point. So in summary the (c, S, α) policy works as follows. After a new demand has occurred, register the new vector of inventory positions. Check if (2') is fulfilled and in case of yes, take all items in the sets (3) and (4) and bring their inventory positions to level S_j . Similar to Johansen and Thorstenson (2006), but unlike a standard control policy where we can look-up the decision in a table, we need to perform the check in (2') repeatedly. However, this is computationally very fast to do, so it is not a hindrance for an implementation of the procedure. In finalizing this section, note also, that the determination of when to make a joint order, by using (2'), is determined from the summation of cost savings across items. This is contrary to the common can-order policy with item-dependent re-order points that

determines when to make a joint order. Thus, our policy is significantly different from a standard can-order policy by not having these must order points, at all.

4. Mathematical derivation of the one-step policy-iteration procedure

In this section we outline how to derive the function $R_j()$ that is the input to (2'). Following the description in Section 1, we assume that the demand for each item is given by independent Poisson processes, each with rates λ_j , $j=1, \dots, N$. Let us assume that the inventory control of each item is operated separately, and that $\bar{K}_j$ is the assigned order cost. Let $g_j(y)$ be the expected inventory and backorder costs of item j , if its inventory position is kept constant at level y . This is, see also Zipkin (2000; Chapter 6), specified as

$$g_j(y) = (h_j + p_j) \sum_{d=0}^{y-1} (y-d) P(D_j(L_j) = d) + \pi_j \lambda_j P(D_j(L_j) \geq y) + p_j (\lambda_j L_j - y) \quad (5)$$

Here it is understood that the summation term vanishes if $y \leq 0$. The random variable $D_j(L_j)$ is aggregate demand of item j during a lead-time period of duration L_j . In this stand-alone case the optimal policy is of type (s, S) , and because of the Poisson demand process, the inventory positions are uniformly distributed on $s_j + 1, \dots, S_j$. Hence the relevant cost expression to minimize is

$$C_j(s_j, S_j) = \frac{\lambda_j \bar{K}_j + \sum_{y=s_j+1}^{S_j} g_j(y)}{S_j - s_j} \quad (6)$$

This can be solved efficiently by the algorithm of Federgruen and Zheng (1992). Let (s_j^*, S_j^*) be the optimal policy and let C_j^* be the expected costs per time unit of the optimal policy. From the specifications of the algorithm the following optimality conditions can be derived

$$S_j^* = \text{Max}\{y: g_j(y) < C_j^*\} \quad (7)$$

and

$$s_j^* = \text{Min}\{y: g_j(y+1) \leq C_j^*\} \quad (8)$$

Therefore if we have a desired order-up-to value $\bar{S}_j$ for item j we can make this choice optimal in the optimization of (6) by choosing a desired minimum cost value $\bar{C}_j$ in the interval $(g_j(\bar{S}_j), g_j(\bar{S}_j + 1)]$. Set $\bar{s}_j$ as

$$\bar{s}_j = \text{Min}\{y: g_j(y+1) \leq \bar{C}_j\} \quad (9)$$

and compute $\bar{K}_j$ as

$$\bar{K}_j = \frac{1}{\lambda_j} \sum_{y=\bar{s}_j+1}^{\bar{S}_j} (\bar{C}_j - g_j(y)) \quad (10)$$

Then we get that when the order cost is $\bar{K}_j$, the optimal pair of (s, S) is $(\bar{s}_j, \bar{S}_j)$ with minimum cost $\bar{C}_j = C_j(\bar{s}_j, \bar{S}_j)$. The associated relative values are, see Tijms (1994; Section 3.6.1)

$$v_j(y) = \begin{cases} \bar{K}_j + \frac{1}{\lambda_j} \sum_{r=\bar{s}_j+1}^y g_j(r) - \frac{1}{\lambda_j} (y - \bar{s}_j) \bar{C}_j & y > \bar{s}_j \\ \bar{K}_j & y \leq \bar{s}_j \end{cases} \quad (11)$$

It is the expected costs incurred for item j from an initial state y until we reach the next order instant minus the expected time until the next order instant multiplied with the minimum cost rate. It can be rewritten as

$$v_j(y) = \begin{cases} [C_j(\bar{s}_j, y) - C_j(\bar{s}_j, \bar{s}_j)] \frac{y - \bar{s}_j}{\lambda_j} & y > \bar{s}_j \\ \bar{K}_j & y \leq \bar{s}_j \end{cases} \quad (12)$$

This specification makes it apparent that $v_j(y) \geq 0$ for any y and that $v_j(\bar{s}_j) = 0$. Assume now at current state $(x_1, \dots, x_N)$ that we would deviate from the stand-alone decision rules and make a joint order decision, altering the inventory positions to $(a_1, \dots, a_N)$, with $a_j \geq x_j$ for $j = 1, \dots, N$, but otherwise immediately thereafter (at the next demand epoch for any item) we go back to the stand-alone rules. The following function $R_j(a_j)$ measures the aggregated costs, assuming the stand-alone universe, in excess of the target $\bar{C}_j$ until a new order following the $(\bar{s}_j, \bar{S}_j)$ policy is made

$$R_j(a_j) = \begin{cases} v_j(a_j) & a_j > \bar{s}_j \\ \frac{g_j(a_j) - \bar{C}_j}{\lambda_j} + v_j(a_j) & a_j \leq \bar{s}_j \end{cases} \quad (13)$$

In the upper part we are immediately transferred to a recurrent state within the regime of the stand-alone $(\bar{s}_j, \bar{S}_j)$ policy. This explains the term $v_j(a_j)$. When $a_j \leq \bar{s}_j$, we are bound to stay in a transient state of the stand-alone $(\bar{s}_j, \bar{S}_j)$ policy for an average duration of $1/\lambda_j$ before an action following the $(\bar{s}_j, \bar{S}_j)$ policy can be made, and this action is an order-up-to decision, and from the lower part of (12) we get $v_j(a_j) = \bar{K}_j$. Therefore when $a_j \leq \bar{s}_j$ we can rewrite $R_j(a_j)$ in (13) to $\frac{C_j(a_j - 1, a_j) - \bar{C}_j}{\lambda}$ which is non-negative. Therefore we can conclude that an unconstrained minimum of the function $R_j(a_j)$ is attained for $a_j = \bar{s}_j$ with $R_j(\bar{s}_j) = 0$.

5. Development of the simulation model

For collecting the average long-run cost of the (c, S, α) policy we should in principle compute the steady-state probability distribution $q(x_1, \dots, x_N)$ of any vector of inventory positions $(x_1, \dots, x_N)$. These should then be used as weights for corresponding costs of making an action $(a_1, \dots, a_N)$ from (2'), (3) and (4). Specifically this means that the expected long-run average cost of the (c, S, α) policy should be computed as:

$$\sum_{(x_1, \dots, x_N)} q(x_1, \dots, x_N) \left[\mu K I_{\{\exists j: a_j > x_j\}} + \sum_{j=1}^N \left[\mu k_j I_{\{a_j > x_j\}} + g_j(a_j) \right] \right] \quad (14)$$

where the vector $(a_1, \dots, a_N)$ is found from (2'), (3) and (4). Note that we here multiply the order costs with $\mu = \sum_{j=1}^N \lambda_j$, the aggregate demand rate across all items, to take into account that the expected

time until next demand occurrence is $1/\mu$. As it would be numerically very hard to determine the steady state probability distribution, we estimate (14) through a simulation model developed in Visual Basic for Applications (VBA). All simulated average costs have 95% confidence intervals computed from 20 replications. Each replication simulates the inventory system over 100000 demand occurrences following 2000 demand instances used for warming-up the system. We set

$$\bar{c}_j = \frac{g_j(s_j) + g_j(s_{j+1})}{2} \quad j=1, \dots, N \quad (15)$$

Each replication always starts with the inventory positions at their order-up-to levels. During a simulation we monitor how the inventory system evolves according to the (c, S, α) policy and collect the costs of the system, as outlined. Simultaneously we monitor how the inventory system will evolve if following the $Q(s, S)$ policy and collects the average costs for this as well. As the two inventory control policies in this way are exposed to the same stochastic stimulus, and the cost performances are positively correlated, it is possible in each replication to collect the difference in average costs between these and develop a paired- t confidence interval, Law and Kelton (1991; Section 10.2.1).

6. Numerical studies

We initially investigate the standard 12 item problem, Table 1 in Atkins and Iyugun (1988), which is the dataset analyzed in almost all numerical experiments on the stochastic joint replenishment problem. The optimal $Q(s, S)$ policy for this dataset is reported in Table 1 of Nielsen and Larsen (2005). When using $(S_1, \dots, S_{12})$ as the order-up-to values and $(c_1, \dots, c_{12})$ as the re-order points from this computed policy, respectively and varying α in the interval from 1 to 1.3 in increments of 0.05, we get the following results.

<Table 2 about here>

The optimal $\alpha = 1.2$ and the expected cost are slightly, but not significantly, less than the costs of the optimal $Q(s, S)$ policy. So we cannot beat the $Q(s, S)$ policy but we are statistically equally good as the $Q(s, S)$ policy. The expected costs are 2233.3 when the method of Johansen and Thorstenson (2006) is applied. However, they use a much more numerically extensive method.

Now we examine datasets with heterogeneity among the items. We conduct 4 experiments in all, and denote them *ExpHet_1*, *ExpHet_2*, *ExpHet_3* and *ExpHet_4*. In each experiment we have 12 items in all and we have $n = 1, 2, 3, 4$ items which we denote ‘accessories’ which have high demand but low value. The remaining items, in $12 - n$, are denoted ‘main components’ and are characterized by having low demand and high value. In order to simplify the presentation, all ‘accessories’ are identical with regard to their parameters (demand and cost parameters). The same holds for the ‘main components’. The datasets have been developed with inspiration from Brouer (2009). The dataset *ExpHet_2* is developed from the dataset of *ExpHet_1* by scaling the cost components h and p by a factor 0.5. Similarly the dataset *ExpHet_3* is developed from the dataset of *ExpHet_1* by scaling the cost components h and p by a factor 2. Finally the dataset *ExpHet_4* is developed from the dataset of *ExpHet_1* by making the ‘main component’ even more expensive and by lowering the joint order cost to 50. In each experiment, for each $n = 1, 2, 3, 4$, we first compute the optimal $Q(s, S)$ policy. Then we select the optimal α by running our simulation models for different α values over a selected interval. When having found the optimal α we carry out a detailed simulation of the (c, S, α) policy as

well as for the corresponding $Q(s,S)$ policy. We can then report the difference in performance between these policies. In addition to this, we also report on the performance of the independent solution in order to verify that it makes sense to apply a coordinated inventory policy for this particular dataset. It turns out for the first three datasets, as n reaches the level of 4, the improvement of the (c,S,α) policy compared to the optimal $Q(s,S)$ policy declines. However, for the fourth dataset there is still a substantial improvement potential of the (c,S,α) policy compared to the optimal $Q(s,S)$ policy when n is 4. Subsequently, for dataset *ExpHet_4* we extend the experiment with up to 8 ‘accessory’ items (Table 6c). The results of our numerical experiments are stated in Tables 3a – 6c.

<Tables 3a – 6c about here>

In general the numerical experiments demonstrate significant improvements of the (c,S,α) policy compared to the optimal $Q(s,S)$ policy. Furthermore, it can be sensible to use the (c,S,α) policy as the mechanism of $Q(s,S)$ policy anyway might be difficult to explain to a practitioner.

We will finalize this section by reporting the performance of (c,S,α) policy on the dataset of Dance et al. (2012) presented in Table 1. Using the computed $Q(s,S)$ policy for this dataset, stated in Section 2, and by varying the parameter α from 1.55 to 2.10 in steps of 0.05, we get for all those cases a simulated cost rate of 36.64 (the half length of 95% confidence interval is 0.01 in all the cases). So for this (maybe extreme) dataset we can make a very significant improvement of the (inferior) $Q(s,S)$ policy. The results are almost as good as those reported by Dance et al. (2012) for their policy.

7. Comparisons to the can-order policy

Melchioris (2002) has through extensive simulations found the optimal can-order policies for 6 datasets, all with identical items. These are reported in Table 7. Note that for dataset no. 6, the optimal can-order policy is better than the optimal $Q(s,S)$ policy. We adjust each can-order policy to a (c,S,α) policy for which the only remaining issue is to find an optimal α . As can be seen from Table 7 we improve the given can-order policy for all 6 datasets. The least improvement is in the last dataset (where the can-order policy is better than the $Q(s,S)$ policy) and where, somewhat unexpectedly, the optimal α parameter is also much larger than in all other instances. As a comparison we have also stated the corresponding optimal $Q(s,S)$ policies and the result when adapting them to (c,S,α) policies, by finding optimal α parameters. Again we see, as in Table 2, that the (c,S,α) policy perform equally well as the $Q(s,S)$ policy.

<Table 7 about here >

8. Concluding remarks

In this paper, we have shown how to develop a new policy, mimicking the policy iteration step in the policy-iteration algorithm, for the stochastic joint replenishment problem. The numerical results shows it can improve an optimal $Q(s,S)$ policy in the case where there is heterogeneity among the items. Furthermore our results also indicate the new policy can improve a given optimal can-order policy. The method described in the paper can be interpreted as a way to tackle the ‘curse of dimensionality’ by making a decomposition of a value function that is in principle of dimension N , into a sum of value functions of dimension I . Our work indicates that such decomposition, although approximate, will produce reasonably good results.

References

- Atkins D.R., Iyogun P.O., 1988. Periodic versus “can order” policies for coordinated multi-item inventory systems. *Management Science*, 34, 791-796.
- Axsäter S., 2003. A new decision rule for lateral transshipments rule in inventory systems. *Management Science*, 49, 1168-1179.
- Axsäter S., 2007. A heuristic for triggering emergency orders in an inventory system. *European Journal of Operational Research*, 176, 880-891.
- Balintfy J.L., 1964. On a class of multi-item inventory problems. *Management Science*, 10, 287-297.
- Brouer T., 2009. Analyse af tre udvalgte JRP politikker (in Danish). Master Thesis in Mathematical Economics, Aarhus University, Denmark.
- Dance C.R., Zoeter O.R., Lee H., 2012. Simple policies for joint replenishment can perform badly. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, March 2012, Hong Kong, http://www.iaeng.org/publication/IMECS2012/IMECS2012_pp1455-1460.pdf
- Federgruen A. Groenevelt H., Tijms, H., 1984. Coordinated replenishments in a multi-item inventory system with compound Poisson demands. *Management Science*, 30, 344-357.
- Federgruen A, Zheng Y.S., 1992. An efficient algorithm for computing an optimal (R,Q) policy in continuous review stochastic inventory systems. *Operations Research*, 40, 808-813.
- Johansen S.G., Thorstenson A., 2006. A Value based Q(s,S) for joint replenishments. Working Paper, Department of Business Studies, Aarhus University, Denmark, http://pure.au.dk/portal/files/32352761/VQ_EJOR.pdf
- Johansen S.G., Melchior P., 2003. The can-order policy for the periodic review joint replenishment problem. *Journal of the Operational Research Society*, 54, 283-290.
- Larsen C., 2008. Comments to Özkaya et al. (2006). Working Paper L-2008-04, Department of Business Studies, Aarhus School of Business, Denmark, http://pure.au.dk/portal/files/2469/L_2008_04.PDF
- Larsen C., 2009. The Q(s,S) policy for the joint replenishment problem extended to the case of correlation among item-demands. *International Journal of Production Economics*, 118, 292-297.
- Larsen C., 2011. To the editor. *Naval Research Logistics*, 58, 410.
- Law A.M., Kelton W.D., 1992. *Simulation modelling & analysis*, 2nd. ed. McGraw-Hill, New York, NY, USA.
- Melchior P., 2002. Calculating can-order policies for the joint replenishment problem by the compensation approach. *European Journal of Operational Research*, 141, 587-595.
- Nielsen C.S., Larsen C., 2005. An analytical study of the Q(s,S) policy applied to the joint replenishment problem. *European Journal of Operational Research*, 163, 721-732.
- Özkaya B.Ö., Gürlü Ü., Berk E., 2006. The stochastic joint replenishment problem: a new policy, analysis and insights. *Naval Research Logistics*, 53, 525-546.
- Pantumsinchai P., 1992. A comparison of three joint ordering inventory policies. *Decision Sciences*, 23, 111-127.
- Powell W.B., 2011. *Approximate dynamic programming – solving the curse of dimensionality*. John Wiley, Hoboken, NJ, USA.
- Renberg B., Planche R., 1969. Un modèle pour la gestion simultanée des n articles d'un stock. *Revue Française d'Informatique et de Recherche Opérationnelle*, 6, 47-59.
- Schultz H, Johansen S.G., 1999. Can-order policies for coordinated inventory replenishment with Erlang distributed times between ordering. *European Journal of Operational Research*, 113, 30-41.
- Silver E.A., 1981. Establishing reorder points in the (S,c,s) coordinated control system under a compound Poisson demand. *International Journal of Production Research*, 19, 743-750.

- Tijms H., 1994. Stochastic models: An Algorithmic Approach. John Wiley, New York, NY.
- Tijms H., 2003. A First Course in Stochastic models. John Wiley, New York, NY.
- Viswanathan S., 1997. Periodic review (s,S) policies for joint replenishment inventory systems. Management Science, 43, 1447-1454.
- Zheng Y.S., Federgruen A., 1991. Finding optimal (s,S) policies is just as simple as evaluating a single policy. Operations Research, 39, 654-665.
- Zipkin P.H., 2000. Foundations of inventory management, McGraw Hill, Boston, MA, USA.

Tables and figures

	Items 1 and 2	Items 3 and 4
Demand rate λ	4	1
Inventory cost h	1	125
Penalty cost π	0	0
Penalty cost p	1	125
Item specific order cost k	0.5	0.5
Lead time L	0.1	0.1

Table 1: Dataset number 5 of Dance et al (2012). The joint order cost K is 3.

α	The difference between the simulated costs of $Q(s,S)$ policy and the (c,S,α) policy	Average costs of the (c,S,α) policy
1	-11.43 +/- 2.19	2263.77 +/- 2.17
1.05	-6.30 +/- 2.29	2258.65 +/- 2.12
1.1	-1.72 +/- 1.67	2254.07 +/- 1.47
1.15	-1.33 +/- 1.46	2253.68 +/- 1.18
1.2	0.32 +/- 1.53	2252.03 +/- 1.60
1.25	-3.39 +/- 1.41	2255.73 +/- 1.51
1.30	-5.50 +/- 1.28	2257.85 +/- 1.29

Table 2: Comparison between the (c,S,α) policy and the $Q(s,S)$ policy for the standard 12 item problem of Atkins and Iyogun (1988) Table 1. The simulated costs of the optimal $Q(s,S)$ policy are 2252.34 +/- 0.94. The optimal exact average costs of the $Q(s,S)$ policy are 2251.50.

	"Accessory"	"Main components"
Demand rate λ	10	1
Inventory cost h	0.1	2
Penalty cost π	0	0
Penalty cost p	2	30
Item specific order cost k	10	10
Lead time L	1	1

Table 3a: Specification the data for *ExpHet_1*. The joint order cost $K = 100$.

	"Accessory"	"Main components"
Demand rate λ	10	1
Inventory cost h	0.05	1
Penalty cost π	0	0
Penalty cost p	1	15
Item specific order cost k	10	10
Lead time L	1	1

Table 4a: Specification the data for *ExpHet_2*. The joint order cost $K = 100$.

	"Accessory"	"Main components"
Demand rate λ	10	1
Inventory cost h	0.2	4
Penalty cost π	0	0
Penalty cost p	4	60
Item specific order cost k	10	10
Lead time L	1	1

Table 5a: Specification the data for *ExpHet_3*. The joint order cost $K = 100$.

	"Accessory"	"Main components"
Demand rate λ	10	1
Inventory cost h	0.1	8
Penalty cost π	0	0
Penalty cost p	2	120
Item specific order cost k	10	10
Lead time L	1	1

Table 6a: Specification the data for *ExpHet_4*. The joint order cost $K = 50$.

# 'accessories'	1	2	3	4
# 'main components'	11	10	9	8
Optimal $Q(s,S)$ policy:				
Q	73	106	169	209
s_{ac}	33	34	41	42
S_{ac}	45	46	54	54
s_{mc}	4	4	5	5
S_{mc}	6	6	7	7
cost rate	158.28	152.66	146.38	139.49
Optimal α	1.00	1.10	0.80	0.80
(c,S,α) policy	153.64 +/- 0.14	148.02 +/- 0.18	142.97 +/- 0.18	136.98 +/- 0.20
$Q(s,S)$ policy	158.25 +/- 0.18	152.55 +/- 0.22	146.56 +/- 0.27	139.52 +/- 0.20
Difference	4.61 +/- 0.22	4.53 +/- 0.13	3.58 +/- 0.26	2.54 +/- 0.16
Difference in %	2.91%	2.97%	2.44%	1.82%
Independent solution	250.49	243.58	236.67	229.76

Table 3b: Results for *ExpHet_1*.

# 'accessories'	1	2	3	4
# 'main components'	11	10	9	8
Optimal $Q(s,S)$ policy:				
Q	114	163	215	305
s_{ac}	47	48	48	56
S_{ac}	64	64	65	73
s_{mc}	5	5	5	6
S_{mc}	8	8	8	9
cost rate	102.66	98.67	94.92	90.82
Optimal α	0.95	1.05	1.15	0.85
(c,S,α) policy	99.82 +/- 0.13	96.35 +/- 0.13	92.71 +/- 0.15	89.51 +/- 0.14
$Q(s,S)$ policy	102.28 +/- 0.12	98.59 +/- 0.17	94.96 +/- 0.21	90.88 +/- 0.16
Difference	2.47 +/- 0.14	2.24 +/- 0.15	2.24 +/- 0.18	1.37 +/- 0.15
Difference in %	2.41%	2.27%	2.35%	1.51%
Independent solution	173.83	169.22	164.61	160.00

Table 4b: Results for *ExpHet_2*.

# 'accessories'	1	2	3	4
# 'main components'	11	10	9	8
Optimal $Q(s,S)$ policy:				
Q	47	70	93	117
s_{ac}	24	25	26	26
S_{ac}	54	35	36	36
s_{mc}	3	3	3	3
S_{mc}	5	5	5	5
cost rate	248.86	239.42	229.36	218.92
Optimal α	0.85	0.90	1.00	1.10
(c,S,α) policy	239.86 +/- 0.16	231.39 +/- 0.15	221.98 +/- 0.24	213.78 +/- 0.26
$Q(s,S)$ policy	248.93 +/- 0.19	239.61 +/- 0.23	229.47 +/- 0.35	218.79 +/- 0.37
Difference	9.07 +/- 0.21	8.22 +/- 0.21	7.49 +/- 0.36	5.01 +/- 0.32
Difference in %	3.64%	3.43%	3.26%	2.29%
Independent solution	369.91	358.82	347.73	336.64

Table 5b: Results for *ExpHet_3*.

# 'accessories'	1	2	3	4
# 'main components'	11	10	9	8
Optimal $Q(s,S)$ policy:				
Q	22	32	43	54
s_{ac}	14	14	14	14
S_{ac}	56	56	57	57
s_{mc}	2	2	2	2
S_{mc}	4	4	4	4
cost rate	359.87	337.80	314.96	291.68
Optimal α	0.50	0.60	0.70	0.70
(c,S,α) policy	344.93 +/- 0.09	321.18 +/- 0.21	298.21 +/- 0.15	274.57 +/- 0.17
$Q(s,S)$ policy	359.87 +/- 0.19	337.85 +/- 0.22	315.03 +/- 0.24	291.34 +/- 0.35
Difference	15.44 +/- 0.19	16.67 +/- 0.25	16.82 +/- 0.26	16.76 +/- 0.30
Difference in %	4.29%	4.93%	5.34%	5.75%
Independent solution	442.20	413.76	402.98	392.20

Table 6b: Results for *ExpHet_4*.

# 'accessories'	5	6	7	8
# 'main components'	7	6	5	4
Optimal $Q(s,S)$ policy:				
Q	66	80	95	170
s_{ac}	15	15	15	21
S_{ac}	57	58	58	52
s_{mc}	2	2	2	3
S_{mc}	4	4	4	5
cost rate	268.06	244.13	219.89	193.42
Optimal α	0.70	0.75	0.80	0.65
(c,S,α) policy	251.12 +/- 0.24	228.00 +/- 0.24	205.06 +/- 0.28	186.55 +/- 0.33
$Q(s,S)$ policy	267.75 +/- 0.41	243.96 +/- 0.40	219.90 +/- 0.51	193.04 +/- 0.52
Difference	16.63 +/- 0.30	15.95 +/- 0.36	14.85 +/- 0.45	6.49 +/- 0.53
Difference in %	6.21%	6.54%	6.75%	3.36%
Independent solution	381.42	370.64	359.86	349.08

Table 6c: Results for *ExpHet_4*, extended.

Figure 1: The algorithm in schematic form

Data set	Optimal (s,c,S) policy				Corresponding (c,S,α) policy				Optimal $Q(s,S)$ policy				Corresponding (c,S,α) policy			
	s	c	S	cost	c	S	α	cost	Q	s	S	cost	c	S	α	cost
1	7	27	28	1405	27	28	0.95	1395.62 +/- 0.80	178	25	28	1393.72	25	28	0.95	1395.66 +/- 0.79
2	5	26	31	1698	26	31	1.15	1681.69 +/- 0.76	226	25	31	1680.12	25	31	1.15	1682.05 +/- 0.82
3	8	23	24	1102	23	24	0.9	1092.31 +/- 0.48	121	20	24	1091.98	20	24	0.9	1091.90 +/- 0.56
4	6	26	29	1477	26	29	0.75	1463.04 +/- 0.75	198	22	29	1463.00	22	29	0.75	1462.55 +/- 0.62
5	10	18	23	981	18	23	0.55	979.60 +/- 0.40	102	19	23	980.34	19	23	0.55	978.73 +/- 0.36
6	9	15	27	1377	15	27	2,8	1375.91 +/- 0.42	76	15	28	1390.40	15	28	1.1	1380.65 +/- 0.43

Table 7: The data set of Melchior (2002), Table 1. In all datasets $N = 12$ and the items are identical (why subscripts are suppressed) with $h = 6$, $\pi = 30$, $\lambda = 10$ and $L = 1$. In the first 2 data sets $p = 10$ and otherwise 0. The joint order cost $K = 500$ in data sets 1 and 2, 150 in data set 3 and 4 and 50 in data set 5 and 6. In the unevenly numbered data set $k = 10$ and 50 otherwise. The results of the $Q(s,S)$ policy can also be seen in Brønmo (2005) at p. 96, though in there a few minor inaccuracies appear.