Journal of the Operational Research Society (1999) 50, 789–796 0160-5682/99 $12.00
http://www.stockton-press.co.uk/jor

A fast heuristic for minimising total average cycle stock subject to practical constraints

EA Silver[1] and I Moon[2]
[1]*The University of Calgary, Canada and* [2]*Pusan National University, Korea*

We address a problem of setting reorder intervals (time supplies) of a population of items, subject to a restricted set of possible intervals as well as a limit on the total number of replenishments per unit time, both important practical constraints. We provide a dynamic programming formulation for obtaining the optimal solution. In addition, a simple and efficient heuristic algorithm has been developed. Computational experiments show that the performance of the heuristic is excellent based on a set of realistic examples.

Keywords: inventory; cycle stock; dynamic programming; heuristic

Introduction

Inventory managers are very concerned with the efficient use of constrained resources. One example is the minimisation of the total number (or cost) of replenishments per unit time across a population of items given a total budget that can be allocated among the average inventories of the items. Conversely, how does one keep the total average stock (in monetary units) as low as possible subject to a restriction on the total number of replenishments per year (that is, for a given workload capability of the purchasing or production department)? Under economic order quantity assumptions the optimal solutions to these constrained problems have been known for many years. In fact, by varying the level of the constrained resource, one can trace out a whole exchange curve which shows the best that one can do on one aggregate measure as the constraint level of the other aggregate measure is varied.[1–4] However, these results hinge upon continuous possible values of the decision variables, for example the order quantities. In practice, managers often prefer to restrict the decision variables to a set of discrete possible values, for example the order quantity of any item is restricted to a given set of time supplies such as 1 week, 2 weeks, 1 month, 2 months, 3 months, 6 months, or 1 year. This facilitates easier understanding and implementation. Two related questions are: (i) is it easy to incorporate the restriction of a discrete set of possible values of the time supply?; and (ii) what is the cost of degradation associated with adding this practical type of constraint?

Correspondence: Dr EA Silver, Faculty of Management, The University of Calgary, Calgary, Alberta, Canada T2N 1N4.
Email: silver@mgmt.ucalgary.ca

This paper addresses how to deal with the above constrained problem in a pragmatic fashion. Specifically we treat the case where there is a set of time supplies, one of which must be used for each of a population of items. There is a specified upper limit on the aggregate number of replenishments per year and we wish to choose the time supply order quantities of the population of items, subject to the discrete set of options and the aggregate constraint, so as to minimise the monetary value of the total average stock level (equivalently, to minimise the total annual carrying charges).

In the next section we introduce the notation and mathematically formulate the problem, including obtaining a useful lower bound. This is followed by the specification of an optimal solution procedure using dynamic programming. The associated computational effort required increases substantially with the number of items involved and with the total number of replenishments permitted. Perhaps more important, dynamic programming is a difficult concept to explain to practitioners. Therefore, we subsequently present a heuristic approach that overcomes both of these drawbacks. Moreover, we present results of computational experiments that show that very little, if any, cost degradation occurs through the use of the heuristics. The paper concludes with a summary section.

Problem formulation and a lower bound

The notation to be used is as follows:

n = number of items
m = number of possible discrete reorder intervals
N = maximum total number of replenishments per unit time

D_i = the demand rate of item i, in units/unit time, $i = 1, \ldots, n$

v_i = the unit variable cost of item i, in monetary units/unit, $i = 1, \ldots, n$

$w_i \equiv \frac{D_i v_i}{2}$, $i = 1, \ldots, n$

t_i = the reorder interval of item i, $i = 1, \ldots, n$ (these are the decision variables)

T = the set of possible reorder intervals

T_j = the jth possible reorder interval, $j = 1, \ldots, m$

y_i = the number of orders for item i per unit time, $i = 1, \ldots, n$ (alternate set of decision variables)

Y = the set of possible numbers of orders per unit time

Y_j = the jth possible (in increasing order) number of orders per unit time, $j = 1, \ldots, m$

The assumptions are exactly the same as for the classical multi-item economic order quantity model.[5] Without loss of generality, we assume that the items are numbered such that $w_1 \leq w_2 \leq \cdots \leq w_n$.

Then our problem can be represented as follows:

(P1)
$$\min \sum_{i=1}^{n} w_i t_i$$

subject to

$$\sum_{i=1}^{n} \frac{1}{t_i} \leq N \tag{1}$$

$$t_i \in T = \{T_1, \ldots, T_m\} \ \forall i \tag{2}$$

The objective function minimises the total average cycle stock. Constraint (1) restricts the total number of replenishments per unit time. In order to facilitate obtaining an optimal solution we transform **(P1)** into the following equivalent problem **(P2)**.

(P2)
$$\min \sum_{i=1}^{n} \frac{w_i}{y_i}$$

subject to

$$\sum_{i=1}^{n} y_i \leq N \tag{3}$$

$$y_i \in Y = \{Y_1, Y_2, \ldots, Y_m\} \ \forall i \tag{4}$$

where the Y_js are positive integers.

We can always ensure that the Y_js are all integers by redefining the time unit as the least common denominator of the set of $1/\tau_j$s as will be illustrated later in a numerical example.

Numerical Illustration

A common set of reorder intervals in practice is as follows:

T = {1 week, 2 weeks, 3 weeks, 1 month, 2 months, 3 months, 4 months, 6 months, 12 months}

Then our problem can be represented as follows:

(P1)
$$\min \sum_{i=1}^{n} w_i t_i$$

subject to

$$\sum_{i=1}^{n} \frac{1}{t_i} \leq N$$

$t_i \in$ {1 week, 2 weeks, 3 weeks, 1 month, 2 months, 3 months, 4 months, 6 months, 12 months} $\forall i$

The reciprocals are (on an annual basis):

$$\frac{1}{t_i} \in \{52, 26, \tfrac{52}{3}, 12, 6, 4, 3, 2, 1, \}$$

Therefore, the above problem is equivalent to the following problem where we use three years as the basic time unit:

(P2)
$$\min \sum_{i=1}^{n} \frac{3w_i}{y_i}$$

subject to

$$\sum_{i=1}^{n} y_i \leq 3N$$

$$y_i \in \{3, 6, 9, 12, 18, 36, 52, 78, 156\} \ \forall i$$

If we solve **(P2)**, it gives a solution for the number of orders per three years for each item. We can transform the solution into the selection of the reorder intervals. For example, suppose $y_i = 52$, which means we need to order 52 times per three years. Then, the reorder interval will be $t_i = \frac{3}{52}$ year or 3 weeks.

Lower bound on (P2)

We derive a lower bound on **(P2)** for two reasons. Firstly, it represents the optimal value of the objective function when the t_is are not restricted to a discrete set of values. Therefore, it will provide an indication of the degradation (increase in the total average stock value) caused by the introduction of the pragmatic constraint of restricting the t_is to the discrete set T. The second reason is that the lower bound solution is used as a starting point in the heuristic.

If we relax constraint (4), we get a relaxed version of **(P2)** as follows.

(LB)
$$\min \sum_{i=1}^{n} \frac{w_i}{y_i}$$

subject to

$$\sum_{i=1}^{n} y_i \leqslant N$$

If we can find a Kuhn–Tucker (KT) solution,[6] it will be a global minimum due to the fact that the objective function is convex and constraint (3) is a convex set. The Lagrangian function is as follows:

$$L(y_1, \ldots, y_n, \lambda) = \sum_{i=1}^{n} \frac{w_i}{y_i} + \lambda\left(\sum_{i=1}^{n} y_i - N\right)$$

Then, the KT conditions become as follows:

$$\frac{\partial L}{\partial y_i} = -\frac{w_i}{y_i^2} + \lambda = 0 \forall i \tag{5}$$

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{n} y_i = N \tag{6}$$

If we solve (5) and (6) simultaneously, we can obtain a lower bound for **(P2)** as follows:

$$\lambda = \left(\frac{\sum_{i=1}^{n} \sqrt{w_i}}{N}\right)^2 \tag{7}$$

$$y_i^0 = \frac{N\sqrt{w_i}}{\sum_{j=1}^{n} \sqrt{w_j}} \forall i \tag{8}$$

$$\mathrm{LB} = \frac{\left(\sum_{i=1}^{n} \sqrt{w_i}\right)^2}{N} \tag{9}$$

Finding the optimal solution using dynamic programming

Because of the integer nature of the Y_js we will be able to use dynamic programming[7] to solve **(P2)**, in contrast with **(P1)**. The formulation is quite similar to that of the knapsack problem due to the similarity of constraint (3) to the knapsack constraint.[8] However, both constraint (4) and the nonlinear objective function make the formulation somewhat more complicated than the knapsack problem. Therefore, we strongly conjecture that **(P2)** is also an NP-hard[9] problem since the knapsack problem is an NP-hard problem. For large instances of such problems the computational time to obtain the optimal solution becomes prohibitive. Therefore, there is a need for an effective heuristic solution procedure to be discussed later.

First note that from our assumption of $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_n$, the optimal solution has the following property.

$$y_1^* \leqslant y_2^* \leqslant \cdots \leqslant y_n^* \tag{10}$$

The dynamic programming formulation consists of the following three elements:

(1) Stage i is represented by item i, $i = 1, 2, \ldots, n$.
(2) State x_i at stage i is the total number of orders left to be assigned to stages $i, i+1, \ldots, n$.
(3) Alternative y_i at stage i is the number of orders selected for item i. The value of y_i may be as small as Y_1 and as large as y_{i+1}. Let $Y_k \leqslant qN/n < Y_{k+1}$ where q is the smallest integer number of unit periods needed to ensure that all of the Y_j values are integers. Clearly, the upper limit of y_1 must be Y_{k+1} and the lower limit of y_n must be Y_k due to (10).

Let $f_i(x_i)$ = optimal value of stages $i, i+1, \ldots, n$, given the state x_i. Let $y_i^*(x_i)$ be the optimal solution at stage i given state x_i. The lower limit of x_i is clearly $(n-i+1)Y_1$ since the number of orders for each of the remaining items must be at least Y_1. The upper limit of x_i is $qN-(i-1)Y_1$ since the largest number of the remaining orders at stage i would be in the case where the numbers of orders for each of the items 1 to $i-1$ is exactly Y_1. Then the backward recursive equations are as follows:

$$f_n(x_n) = \min_{y_n \in Y}\left\{\frac{qw_n}{y_n}\right\}, \quad x_n = Y_1, Y_1+1, \ldots, qN-(n-1)Y_1$$

$$f_i(x_i) = \min_{y_i \in Y}\left\{\frac{qw_i}{y_i} + f_{i+1}(x_i - y_i)\right\},$$

$$x_i = (n-i+1)Y_1, (n-i+1)Y_1+1, \ldots, qN-(i-1)Y_1, \quad i = 2, \ldots, n-1$$

$$f_1(qN) = \min_{y_1 \in Y}\left\{\frac{qw_1}{y_1} + f_2(qN - y_1)\right\}$$

Numerical example

We solve **(P2)** using a 48 item problem presented by Brown.[2] The Dv values (sorted in increasing order) are listed in Table 1. The individual D and v values are listed in Brown.[2]

For illustrative purposes we set the maximum number of replenishments per year at 700. The reorder intervals are restricted to T = {1 week, 2 weeks, 3 weeks, 1 month, 2 months, 3 months, 4 months, 6 months, 12 months} which, as shown earlier, implies that the possible numbers of orders per three years are restricted to Y = {3, 6, 9, 12, 18, 36, 52, 78, 156}. Now we find an optimal solution for the above example using dynamic programming. For brevity, we provide a condensed outline of the computations.

Table 1 The Dv values for the example

i	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
D_iv_i	20.04	21.72	37.92	54.12	61.8	81.24	94.2	119.4
i	9	10	11	12	13	14	15	16
D_iv_i	171.6	208.8	415.27	470.23	1212	1393.2	1496.4	1600
i	17	18	19	20	21	22	23	24
D_iv_i	1702.4	1714.5	1870.5	1977.8	2647.12	3143.82	4173	4347.78
i	25	26	27	28	29	30	31	32
D_iv_i	4917	5048.3	5397.2	6692.4	6837.6	8003.1	8449.5	9152
i	33	34	35	36	37	38	39	40
D_iv_i	13236.3	13970	15327.6	16368	19765	20470.3	23182.2	25026
i	41	42	43	44	45	46	47	48
D_iv_i	31675.6	56734.2	69040.4	75192	97066.5	99803.2	105984	106465

(Stage 48)

$$f_{48}(x_{48}) = \min_{y_{48}}\left\{\frac{3w_{48}}{y_{48}}\right\} = \min_{y_{48}}\left\{\frac{159697.5}{y_{48}}\right\},$$
$$x_{48} = 3, 4, \ldots, 1959$$

(159697.5 has been obtained from $3w_{48} = 3 \times 106465/2$. 1959 has been obtained from $3N - (n-1)Y_1 = 3 \times 700 - 47 \times 3$.)

It is clear that y^*_{48} is the Y_j that is closest to x_{48} but no larger than x_{48}, for example

$$f_{48}(100) = \min_{y_{48}}\left\{\frac{159697.5}{y_{48}}\right\}$$

$$y^*_{48}(100) = 78 \quad \text{and} \quad f_{48}(100) = 2047.40$$

(Stage 47)

$$f_{47}(x_{47}) = \min_{y_{47}}\left\{\frac{158976}{y_{47}} + f_{48}(x_{47} - y_{47})\right\},$$
$$x_{47} = 6, 7, \ldots, 1962$$

For example

$$\begin{aligned} f_{47}(200) = \min\Big\{&\frac{158976}{3} + f_{48}(197), \frac{158976}{6} \\ &+ f_{48}(194), \frac{158976}{9} + f_{48}(191), \frac{158976}{12} \\ &+ f_{48}(188), \frac{158976}{18} + f_{48}(182), \frac{158976}{36} \\ &+ f_{48}(164), \frac{158976}{52} + f_{48}(148), \frac{158976}{78} \\ &+ f_{48}(122), \frac{158976}{156} + f_{48}(44)\Big\} \end{aligned}$$

$$y^*_{47}(200) = 78 \quad \text{and} \quad f_{47}(200) = 4085.56$$

If we proceed in this way, we arrive at the final stage as follows:

(State 1)

$$f_1(2100) = \min_{y_1}\left\{\frac{30.06}{y_1} + f_2(2100 - y_1)\right\}$$

$$y^*_1(2100) = 3 \quad \text{and} \quad f_1(2100) = 15965.85$$

Consequently, $y^*_i = 3$. After backtracking, we find the optimal solution as follows:

$$\begin{aligned} &y^*_1 = 3, y^*_2 = 3, y^*_3 = 3, y^*_4 = 3, y^*_5 = 3, y^*_6 = 3, y^*_7 = 6, \\ &y^*_8 = 6, y^*_9 = 6, y^*_{10} = 6, y^*_{11} = 9, y^*_{12} = 9, y^*_{13} = 18, \\ &y^*_{14} = 18, y^*_{15} = 18, y^*_{16} = 18, y^*_{17} = 18, y^*_{18} = 18, \\ &y^*_{19} = 18, y^*_{20} = 18, y^*_{21} = 18, y^*_{22} = 36, y^*_{23} = 36, \\ &y^*_{24} = 36, y^*_{25} = 36, y^*_{26} = 36, y^*_{27} = 36, y^*_{28} = 36, \\ &y^*_{29} = 36, y^*_{30} = 36, y^*_{31} = 36, y^*_{32} = 36, y^*_{33} = 52, \\ &y^*_{34} = 52, y^*_{35} = 52, y^*_{36} = 52, y^*_{37} = 52, y^*_{38} = 52, \\ &y^*_{39} = 78, y^*_{40} = 78, y^*_{41} = 78, y^*_{42} = 78, y^*_{43} = 78, \\ &y^*_{44} = 156, y^*_{45} = 156, y^*_{46} = 156, y^*_{47} = 156, y^*_{48} = 156 \end{aligned}$$

As indicated above the optimal objective value is 15965.85. If we convert the above solution to t_is, we obtain the solution summarized in Table 2.

Table 2 Optimal solution in terms of reorder intervals

Reorder interval	*Items*
1 week	44–48
2 weeks	39–43
3 weeks	33–38
1 month	22–32
2 months	13–21
4 months	11–12
6 months	7–10
1 year	1–6

Heuristic algorithm

We develop here a heuristic algorithm for (**P2**) since, as mentioned earlier, it would take too much time to solve the dynamic program if n or N becomes quite large. Moreover, the heuristic solution is much easier for a practitioner to understand than is dynamic programming. Note that we can obtain an upper bound directly from the lower bound solution of the y_is. That is, if we round down the y_i^0s to the nearest Y_js, we obtain an upper bound (a feasible solution) immediately. In contrast, the heuristic algorithm is based on the lower bound (infeasible) solution and a marginal (or greedy) allocation algorithm.[10] At each step the algorithm reduces the number of orders for the item with the least increase of the objective value per unit decrease of the number of orders. This is continued until a feasible solution is obtained. Any capacity (number of orders) remaining is filled in a reverse greedy fashion.

Heuristic algorithm

Step 1 Solve the lower bound problem. Let y_i^0 be the lower bound solution for y_i.

Step 2 Suppose $Y_{r_i} < y_i^0 < Y_{r_i+1}$ (If $y_i^0 = Y_{r_i}$ or Y_{r_i+1}, we just leave y_i^0 as it is). Round up all y_i^0s obtained in solving (LB) to the nearest Y_j (This solution will be always infeasible unless $y_i^0 \in Y = \{Y_1, \ldots Y_m\}$. $\forall i$ since the y_i^0s satisfy the constraint as an equality). Let the current value of y_i be Y_{l_i}.

Step 3 We choose the item i such that

$$\text{argmin}_i \left\{ \frac{\dfrac{qw_i}{Y_{l_i-1}} - \dfrac{qw_i}{Y_{l_i}}}{Y_{l_i} - Y_{l_i-1}} \right\} \tag{11}$$

where q is the smallest integer number of unit periods needed to ensure that all of the Y_j values are integers. Then decrease y_i from Y_{l_i} to Y_{l_i-1} and update the current value of y_i to Y_{l_i-1}.

Step 4 If $\sum_{i=1}^n y_i = qN$, stop. We have found a heuristic solution.
If $\sum_{i=1}^n y_i < qN$, go to Step 5.
If $\sum_{i=1}^n y_i > qN$, go to Step 3.

Step 5 Let the current value of y_i be $Y_{l_i} \forall i$ after the above steps. We choose the item i such that

$$\text{argmax}_i \left\{ \frac{\dfrac{qw_i}{Y_{l_i}} - \dfrac{qw_i}{Y_{l_i+1}}}{Y_{l_i+1} - Y_{l_i}} \right\} \tag{12}$$

Then increase y_i from Y_{l_i} to Y_{l_i+1} if $Y_{l_i+1} - Y_{l_i} \le qN - \sum_{i=1}^n y_i$ and update the current value of y_i as Y_{l_i+1}. If not, eliminate item i from consideration in this step. Repeat this step as long as we can satisfy the constraint.

Numerical example

We solve the same 48 item example as earlier, but this time using the heuristic algorithm with q = 3.

Step 1 First we compute the lower bound solution using (7) and (8). The lower bound solution is as follows:

$$\begin{aligned}
&y_1^0 = 2.02, y_2^0 = 2.10, y_3^0 = 2.27, y_4^0 = 3.32, y_5^0 = 3.55,\\
&y_6^0 = 4.07, y_7^0 = 4.38, y_8^0 = 4.93, y_9^0 = 5.91,\\
&y_{10}^0 = 6.52, y_{11}^0 = 9.19, y_{12}^0 = 9.78, y_{13}^0 = 15.70,\\
&y_{14}^0 = 16.83, y_{15}^0 = 17.44, y_{16}^0 = 18.04, y_{17}^0 = 18.61,\\
&y_{18}^0 = 18.67, y_{19}^0 = 19.50, y_{20}^0 = 20.06, y_{21}^0 = 23.20,\\
&y_{22}^0 = 25.29, y_{23}^0 = 29.13, y_{24}^0 = 29.74, y_{25}^0 = 31.62,\\
&y_{26}^0 = 32.04, y_{27}^0 = 33.13, y_{28}^0 = 36.89, y_{29}^0 = 37.29,\\
&y_{30}^0 = 40.34, y_{31}^0 = 41.45, y_{32}^0 = 43.14, y_{33}^0 = 51.88,\\
&y_{34}^0 = 53.30, y_{35}^0 = 55.83, y_{36}^0 = 57.69, y_{37}^0 = 63.40,\\
&y_{38}^0 = 64.52, y_{39}^0 = 68.66, y_{40}^0 = 71.34, y_{41}^0 = 80.26,\\
&y_{42}^0 = 107.41, y_{43}^0 = 118.49, y_{44}^0 = 123.66, y_{45}^0 = 140.50,\\
&y_{46}^0 = 142.47, y_{47}^0 = 146.81, y_{48}^0 = 147.14
\end{aligned}$$

The value of the lower bound is 15489.64. Note that the value of an upper bound (by rounding down the y_i^0s to the nearest Y_js) is 22651.18.

Step 2 We round up the y_i^0s to the nearest Y_js which provides an initial solution for the heuristic. This solution is clearly infeasible (the sum of the y_is is 2670 which is greater than 2100).

$$\begin{aligned}
&y_1 = 3, y_2 = 3, y_3 = 3, y_4 = 6, y_5 = 6, y_6 = 6, y_7 = 6,\\
&y_8 = 6, y_9 = 6, y_{10} = 9, y_{11} = 12, y_{12} = 12, y_{13} = 18,\\
&y_{14} = 18, y_{15} = 18, y_{16} = 36, y_{17} = 36, y_{18} = 36, y_{19} = 36,\\
&y_{20} = 36, y_{20} = 36, y_{21} = 36, y_{22} = 36, y_{23} = 36, y_{24} = 36,\\
&y_{25} = 36, y_{26} = 36, y_{27} = 36, y_{28} = 52, y_{29} = 52,\\
&y_{30} = 52, y_{31} = 52, y_{32} = 52, y_{33} = 52, y_{34} = 78, y_{35} = 78,\\
&y_{36} = 78, y_{37} = 78, y_{38} = 78, y_{39} = 78, y_{40} = 78,\\
&y_{41} = 156, y_{42} = 156, y_{43} = 156, y_{44} = 156, y_{45} = 156,\\
&y_{46} = 156, y_{47} = 156, y_{48} = 156
\end{aligned}$$

Note that the objective value of the infeasible solution is 12540.95.

Steps 3 and 4 We compute the priority ratios of the reduction of the y values dynamically using (11). The priority order of the reduction of the y values is as follows:

16, 41, 17, 18, 19, 4, 20, 5, 34, 28, 29, 35, 11, 10, 36, 21, 30, 12, 6, 31, 42, 22, 37, 32, 38, 7, 13, 43, 39, 40, ...

It means that we first reduce y_{16} from 36 to 18, and check whether $\sum_i y_i \le 2,100$. We iterate this procedure until we satisfy $\sum_i y_i \leq 2,100$. This finally occurs when we decrease y_{43} from 156 to 78. Since $\sum_i y_i = 2073$, we move to Step 5.

Step 5 Since we have 27 orders remaining, we want to use them up if possible. We compute the priority ratios of the increase of the y values dynamically using (12). The priority order of the increase of the y values is as follows:

$$43, 13, 7, 38, 32, 37, 22, 42, 31, 6, 12, 30, 21, 36, 10, 11,$$
$$35, 29, 28, 34, 5, 33, 9, \ldots$$

Since we can not increase y_{43} from 78 to 156, we increase y_{13} from 12 to 18. Next we increase y_7 from 3 to 6. Next we try to increase y_{38} from 52 to 78, but we have only 18 remaining orders. Therefore, we can not increase y_{38}. Then, we increase y_{32} from 36 to 52. Since we have only two remaining orders, which can not be used to increase any of the y_is, after these assignments, we stop here. Note that the priority order of the increases in the y values is not necessarily in the exact reverse order of that of the Step 3 reduction of the y values (For example, in the above numerical illustration items 20 and 4 appear just before 5 in Step 3 whereas items 33 and 9 appear just after item 5 in Step 5.

The heuristic solution is as follows:

$$y_1 = 3, y_2 = 3, y_3 = 3, y_4 = 3, y_5 = 3, y_6 = 3, y_7 = 6,$$
$$y_8 = 6, y_9 = 6, y_{10} = 6, y_{11} = 9, y_{12} = 9, y_{13} = 18,$$
$$y_{14} = 18, y_{15} = 18, y_{16} = 18, y_{17} = 18, y_{18} = 18, y_{19} = 18,$$
$$y_{20} = 18, y_{21} = 18, y_{22} = 18, y_{23} = 36, y_{24} = 36, y_{25} = 36,$$
$$y_{26} = 36, y_{27} = 36, y_{28} = 36, y_{29} = 36, y_{30} = 36,$$
$$y_{31} = 36, y_{32} = 52, y_{33} = 52, y_{34} = 52, y_{35} = 52, y_{36} = 52,$$
$$y_{37} = 52, y_{38} = 52, y_{39} = 78, y_{40} = 78, y_{41} = 78, y_{42} = 78,$$
$$y_{43} = 78, y_{44} = 156, y_{45} = 156, y_{46} = 156, y_{47} = 156,$$
$$y_{48} = 156$$

The objective value of the heuristic solution becomes 15979.51. Note that this solution is different from the optimal solution in two elements. Specifically, $y_{22} = 18$ and $y_{32} = 52$, compared to $y^*_{22} = 36$ and $y^*_{32} = 36$. Note also that the heuristic solution does not use up the total number of orders. The ratio of the objective value of the heuristic solution to the lower bound is 1.03163, and the ratio of the objective value of the heuristic solution to that of the optimal solution is 1.00086, an extremely small cost penalty.

Computational experiments

First we have solved Brown's 48 item problem varying N to see how the computational time has been affected by N. The program has been coded using GAUSS,[11] and it has been run on an IBM Pentium II PC with 266 Mhz clock speed. The results are shown in Table 3. The computational time to find an optimal solution increases almost linearly. The reason is that the dynamic programming formulation is similar to that of the knapsack problem whose computational complexity is pseudo-polynomial.[9] The performance of the heuristic, compared to the optimal solution, is very encouraging. The penalty of Heu/LB can be regarded as the penalty of restricting the reorder intervals to the finite set with 9 elements. Obviously, if we add some more candidates to the set T, the ratio will be reduced. However, the system control costs (including implementation considerations) increase with the number of options in T.

Table 3 Computational results for Brown's 48 item problem with different values of N

N/year	*LB*	*Optimum*	*Heuristic*	*Heu/Opt*	*Heu/LB*	*CPU time*[a]
100	108 427	119 949	119 949	1.00000	1.10627	1.5
300	36 142	37 499	37 511	1.00032	1.03788	5.6
500	21 685	22 626	22 626	1.00000	1.04339	9.7
700	15 490	15 966	15 980	1.00086	1.03163	14.0
900	12 048	12 387	12 404	1.00137	1.02955	18.1
1100	9857	10 623	10 648	1.00025	1.08025	22.6

[a]CPU time to find the optimal solution (in seconds)

To investigate the performance of the heuristic further, we have tested it using randomly generated problems within realistic parameter ranges. Empirically, it has been found that quite often the distribution of usage values across a population of items can be adequately represented by a lognormal distribution.[1] Consequently, we assume that the Dv values follow a lognormal distribution. If we let Dv be denoted by x then x follows the distribution

$$f(x) = \frac{1}{bx\sqrt{2\pi}} \exp\left[\frac{-(\ln x - a)^2}{2b^2}\right] 0 < x < \infty$$

where a and b are the mean and the standard deviation of the underlying normal distribution. It can be shown that the mean value of the lognormal distribution is given by

$$E(x) = \exp\left[a + \frac{b^2}{2}\right]$$

and the coefficient of variation (a measure of the relative dispersion) of the lognormal distribution is a monotonically increasing function of only b. According to Herron,[12] typically the inventories of merchants (wholesalers, retailers, etc.) have bs in the range of 0.8 to 2.0; industrial producers are in the range 2 to 3; and highly sophisticated hardware suppliers (who are subject to rapid technological innovations) have bs in the 3 to 4 range. Therefore, we set the parameter values within these ranges. Our experiments can be divided into three cases, each in turn having two subcases, one with relatively low variability (denoted as 'dense'), the other with relatively high variability (denoted as 'scattered'). Each subcase involved 15 randomly generated problems. The mean values are in dollars per year.

Case I High Dv values (mean of the Dv values is 11159)

$$n \sim \text{Uniform}(50, 100), \qquad N \sim \text{Uniform}(5n, 10n)$$

$$Dv \sim LN(9, 0.8^2) \text{ for dense set,}$$
$$Dv \sim LN(6, 2.577^2) \text{ for scattered set}$$

Case II Low Dv values (mean of the Dv values is 4105)

$$n \sim \text{Uniform}(50, 100), \qquad N \sim \text{Uniform}(5n, 10n)$$

$$Dv \sim LN(8, 0.8^2) \text{ for dense set,}$$
$$Dv \sim LN(5, 2.577^2) \text{ for scattered set}$$

Case III Low Dv values and small number of items (mean of the Dv values is 4105)

$$n \sim \text{Uniform}(10, 30), \qquad N \sim \text{Uniform}(5n, 10n)$$

$$Dv \sim LN(8, 0.8^2) \text{ for dense set,}$$
$$Dv \sim LN(5, 2.577^2) \text{ for scattered set}$$

The computational results for the above three cases have been summarized in Tables 4, 5 and 6, respectively. Here H, OPT, LB represent the objective value of the heuristic solution, the optimal objective value, and the lower bound value, respectively. The performances of the heuristic for all cases are extraordinarily good. The somewhat higher values of the maximum ratio of H/OPT for the scattered case indicate that the heuristic is not quite as good for relatively high variability of the Dv values. The heuristic works well for both high and low Dv values. The performance deteriorates somewhat as the number of items becomes smaller. For a problem with a small number of items, incorrect assignment of the time supply for an item, especially for an item with a large Dv value, can significantly affect the solution. However, in most practical applications one would expect n values larger than 10 to 30. In fact, the n values could be well above the 50 to 100 range. We did not test above 100 because of the work space limit of the GAUSS software which has been used to find the optimal solution. The H/LB ratios show that restricting the time supplies to a discrete set, instead of permitting any values of the t_is, on average leads to an increase of a few percent in the total average stock level. However, there are some instances, particularly for highly variable Dvs, where the penalty can be considerably higher.

Table 4 Computational results for case I

Set	*Mean (H/OPT)*	*Max (H/OPT)*	*No. OPT*	*Mean (H/LB)*	*Max (H/LB)*
Dense	1.00010	1.00087	8	1.02848	1.03733
Scattered	1.00093	1.00951	5	1.07936	1.48929

Table 5 Computational results for case II

Set	*Mean (H/OPT)*	*Max (H/OPT)*	*No. OPT*	*Mean (H/LB)*	*Max (H/LB)*
Dense	1.00036	1.00199	9	1.03139	1.03954
Scattered	1.00057	1.00705	7	1.04458	1.12984

Table 6 Computational results for case III

Set	*Mean (H/OPT)*	*Max (H/OPT)*	*No. OPT*	*Mean (H/LB)*	*Max (H/LB)*
Dense	1.00585	1.02244	6	1.04675	1.08307
Scattered	1.00543	1.04532	10	1.06623	1.18923

Conclusions

In this paper we have analysed a problem of setting reorder intervals (time supplies) of a population of items, subject to a restricted set of possible intervals as well as a limit on the total number (N) of replenishments per unit time. A dynamic programming formulation for obtaining the optimal solution has been presented. More importantly, a much simpler heuristic solution procedure has been shown to provide excellent results on a set of realistic examples. Therefore we have been able to extend the concept of appropriately allocating stock among items in a way that incorporates the pragmatic consideration of a restricted set of options. Also, by varying the parameter N, one could repeatedly use the heuristic to trace out an exchange curve of total average stock versus N.

Conceptually the same general approach should be applicable to allocating a total available safety stock among a population of items. Work is under way on this safety stock allocation situation and we hope to report useful results in the future.

Acknowledgements—The authors are grateful for the helpful comments made by the anonymous referees. The research leading to this paper was partially carried out while Edward Silver held a visiting Erskine Fellowship in the Department of Management at the University of Canterbury, New Zealand and while Ilkyeong Moon held a visiting position in the Faculty of Management at the University of Calgary during a sabbatical leave. The research has been supported by the Natural Sciences and Engineering Research Council of Canada under Grant A1485, by the Carma Chair at the University of Calgary, and by the LG Yonam Foundation in Korea.

References

1 Starr MK and Miller DW (1962). *Inventory Control: Theory and Practice*. Prentice-Hall: Englewood Cliffs.
2 Brown RG (1967). *Decision Rules for Inventory Management*. Holt, Rinehart and Winston: New York.
3 Girling AJ and Morgan PW (1973). Exchange curves for fixing batch quantities. *Omega* **1**: 241–245.
4 Peterson R and Silver EA (1979). *Decision Systems for Inventory Management and Production Planning and Control*. John Wiley & Sons: New York.

5 Silver EA, Pyke D and Peterson R (1998). *Inventory Management and Production Planning and Scheduling*, 3rd edn. John Wiley & Sons: New York.
6 Luenberger DG (1984). *Linear and Nonlinear Programming*, 2nd edn. Addison-Wesley: Menlo Park.
7 Denardo E (1982). *Dynamic Programming: Theory and Applications*. Prentice-Hall: Englewood Cliffs.
8 Taha HA (1987). *Operations Research*, 4th edn. Macmillan: New York.
9 Garey MR and Johnson DS (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company: San Francisco.
10 Fox B (1966). Discrete optimization via marginal allocation. *Mgmt Sci* **13**: 210–216.
11 Aptech Systems (1997). *GAUSS System Version 3.2.32*. Aptech Systems, Inc.: Maple Valley, Washington.
12 Herron DP (1976). Industrial engineering applications of ABC curves. *AIIE Trans* **8**: 210–218.

Received Ocober 1998;
accepted March 1999 after one revision