An Inventory Replenishment Model for Deteriorating Items with Time-varying Demand and Shortages using Genetic Algorithm

Mohd Omar*, Noor Hasnah Moin, and Ivan Yeo

Institute of Mathematical Sciences, University of Malaya, 50603 Kuala Lumpur, Malaysia.
*E-mail: mohd@um.edu.my

Abstract

In this paper, an inventory replenishment model for deteriorating items is developed. Demand for the item varies with time over a finite planning horizon, during which shortages are allowed and are completely back-ordered. The objective is to determine a replenishment policy that minimizes the total inventory cost. A search procedure based on Genetic Algorithm (GA) is presented and illustrated with some numerical results.

Keywords: Inventory; Replenishment; Time-varying demand; Deterioration; Shortages; Genetic Algorithms

1. Introduction

When developing inventory models, many simplifying assumptions are made to find the optimal replenishment policies. The Economic Order Quantity (EOQ) model assumes a constant demand rate over an infinite planning horizon and minimizes the total inventory cost per unit of time. However, practical demand rates rarely behave this way. Most items experience stable demand only during the saturation phase of their life cycle, while demand is increasing or decreasing during their growth or decline phases respectively. In addition, most models assume that items have infinite shelf-life while in storage. This assumption, while applicable to items with low deterioration rates, seems unrealistic for volatile and radioactive materials, blood banks, food stuff, electronics, etc, that continually lose their utility while in stock. Shortages may be economically desirable in many situations such as when storage costs are high compared to back-order costs or when storage space is limited, is another important aspect that needs to be given special attention.

Hariga (1994) developed an optimal procedure to optimize an inventory model with time-varying demand and shortages. A similar procedure was adopted by Hariga and Benkherouf (1994) to optimize an inventory model with time-varying demand and deterioration, but without shortages. Later, Benkherouf (1995) extended the latter model to include shortages as well. Although this model considered only decreasing demands, it was mentioned that a slight modification would allow the optimal schedule for increasing demands to be found. Other contributors to this model include Wee (1995), Hariga and Al-Alyan (1997), Giri et al. (2000), and Chu and Chen (2002).

Most literatures on this model, as far as the authors are aware of, considered only analytical or heuristical approaches to directly find or approximate its optimal solution. In this paper, we shall consider a stochastic approach by using Genetic Algorithm (GA). Our model operates over a finite planning horizon to satisfy a time-
Mohd Omar, Noor Hasnah Moin, and Ivan Yeo

varying demand and assumes that deterioration occurs at a constant rate and shortages are allowed and are completely back-ordered. The objective is to determine a replenishment policy that minimizes the total inventory cost.

This paper is organised as follows: In the next section, we present the mathematical formulation of our model. In Section 3, we describe a GA-based search procedure to approximate the optimal solution. In Section 4, we illustrate this procedure with some numerical examples. We also compare our results with some optimal and heuristical results. Finally, Section 5 summarizes our findings followed by appendices describing an optimal and several heuristical procedures.

2. Mathematical Formulation

The following assumptions are made in developing our mathematical model:

- A single type of item is held in stock over a finite planning horizon $H$ units of time long.
- Replenishment occurs at an infinite rate with zero lead time and is charged with a cost of $c_1$ per replenishment.
- Inventory holding cost is charged only to good units with a cost of $c_2$ per unit per unit of time.
- Shortages are allowed and are completely back-ordered. Units short are charged with a cost of $c_3$ per unit per unit of time until they are cleared by back-orders.
- The item deteriorates at a fixed rate $\theta$. The deterioration of the units is considered only after their receipt into storage and the deteriorated units are not replaced or repaired during the planning horizon, but are charged with a cost of $c_4$ per unit.
- The demand rate $D(t)$ is a continuous, deterministic and time-varying function and $D(t) > 0$ for $0 \leq t \leq H$.

Suppose that $n$ replenishment (including the final backorder) are made during the planning horizon. Also, suppose that the total time elapsed up to and including the $j$th replenishment cycle ($j = 1, 2, \ldots, n-1$) is given by $t_j$. We define $t_0 = 0$ and $t_{n-1} = H$. Moreover, suppose that the total time elapsed up to the start of shortage in the $j$th cycle is given by $s_j$.

To start with, consider the $j$th cycle. Initially, the stock is zero. Replenishment occurs at $t = t_{j-1}$ and the inventory level reaches its maximum immediately. The positive inventory level is represented by $y_1(t)$. This amount of inventory is depleted over time by demand and deterioration until it reaches zero at $t = s_j$ ($t_{j-1} < s_j < t_j$). Now, a shortage occurs and the negative inventory level is represented by $y_2(t)$. The shortage peaks at $t = t_j$ and is immediately cleared by a back-order. The cycle then repeats itself. We note that the back-order made during the $j$th cycle and the replenishment made during the $(t_{j-1}, t_j)$ cycle incur a single replenishment cost since they both happen at the same time.

Hence, the total inventory cost, which is defined as the sum of replenishment, holding, deterioration and shortage costs, is given by

$$W = nc_1 + \sum_{j=1}^{n-1} \left[ (c_2 + c_4 \theta) \int_{t_{j-1}}^{t_j} y_1(t) \, dt + c_3 \int_{t_{j-1}}^{t_j} y_2(t) \, dt \right].$$

(1)

The variations of $y_1(t)$ and $y_2(t)$ with respect to time in the $(t_{j-1}, t_j)$ cycle is governed by the following linear differential equations:

$$\frac{dy_1(t)}{dt} + \theta y_1(t) = -D(t), \quad t_{j-1} \leq t \leq s_j,$$

(2)
\[
\frac{dy_j(t)}{dt} = -D(t), \quad s_j \leq t \leq t_j,
\]

with the boundary conditions \(y_j(s_j) = 0\) and \(y_j(t_j) = 0\). The solution for equations (2) and (3) is

\[
y_j(t) = \exp(-\theta t) \int_{s_j}^{t} \exp(\theta u) D(u) du, \quad t_{j-1} \leq t \leq s_j,
\]

\[
y_j(t) = -\int_{s_j}^{t} D(u) du, \quad s_j \leq t \leq t_j.
\]

Subsequently, the amount of inventory carried during the \((t_{j-1}, t_j)\) cycle is given by

\[
\int_{s_{j}}^{t_{j}} y_1(t) dt = \theta^{-1} \int_{s_{j}}^{t_{j}} \left( \exp[\theta(t-t_{j-1})] - 1 \right) D(t) dt.
\]

(6)

For the amount of units short during the \((t_{j-1}, t_j)\) cycle, we integrate the positive expression of \(y_2(t)\) over the interval \((s_j, t_j)\):

\[
\int_{s_{j}}^{t_{j}} y_2(t) dt = \int_{s_{j}}^{t_{j}} \left[ \int_{s_{j}}^{t} D(u) du \right] dt.
\]

(7)

It follows that the total inventory cost for an \(n\)-replenishment policy is given by

\[
W = nc_0 + \sum_{j=1}^{n} \left\{ \frac{c_4}{\theta} \left( \int_{s_{j-1}}^{t_{j}} \left( \exp[\theta(t-t_{j-1})] - 1 \right) D(t) dt + \int_{s_{j}}^{t_{j}} D(u) du \right) \right\},
\]

where \(c_a = c_2 + c_4 \theta\).

(8)

Our problem is to minimize \(W\) by finding the optimal number of replenishment, \(n\), and the optimal replenishment points \(t_1, t_2, \ldots, t_{n-1}\), subject to the following constraint:

\[
0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = H,
\]

\(n\) is an integer, \(t_i\) are real numbers.

For a fixed \(n\), the optimal \(s_j\) and \(t_j\) can be obtained from the following set of equations:

\[
\frac{\partial W}{\partial s_j} = 0, \quad j = 1, 2, \ldots, n-1,
\]

\[
\frac{\partial W}{\partial t_j} = 0, \quad j = 1, 2, \ldots, n-2.
\]

Note that, by default, \(t_0 = 0\) and \(t_{n-1} = H\).

As the optimal \(s_j\) is determined in the GA procedure, then \(\frac{\partial W}{\partial s_j} = 0\) gives us

\[
\frac{c_4}{\theta} \left( \exp[\theta(s_j-t_{j-1})] - 1 \right) = t_j - s_j.
\]

(11)

3. Genetic Algorithm (GA) Search Procedure

Genetic Algorithm (GA) is a stochastic search technique that mimics natural biological evolution. GA explores the problem domain by maintaining a population of individuals, each representing a possible solution to the problem, iteratively until some stopping criteria are satisfied. For this type of inventory model, most solution procedures in literature find the optimal or near optimal solution for a fixed \(n\); while the optimal \(n\) is found by evaluating a range of \(W(n)\)'s. However, our GA procedure will consider \(n\) as a variable and evolve it along with the replenishment points.

Each individual in a population encodes the integer value of \(n\) and its associated real-valued replenishment points, \((t_1, t_2, \ldots, t_{n-2})\) that satisfies the following constraints:

\(n \geq 2, \quad 0 = t_0 < t_1 < \cdots < t_{n-2} < t_{n-1} = H\).

Note that the amount of replenishment points is dependant on \(n\), hence not fixed. Consequently, this results in a population of individuals with variable lengths. In order to create the initial population, we randomly
generate a series of \( n \), whose bounds are arbitrarily fixed, and then the appropriate amount of replenishment points for each \( n \). We set the first individual to have \( n = 2 \) and no replenishment points to represent the single replenishment policy with one backorder. Thus, the population array takes the form

\[
\begin{pmatrix}
2 & NaN & NaN & ... & NaN & NaN & NaN \\
\end{pmatrix}
\]

(12)

\[n_1 \quad t_{1,1} \quad t_{1,2} \quad ... \quad t_{1,n_1-1} \quad NaN \quad NaN \]

\[n_2 \quad t_{2,1} \quad t_{2,2} \quad ... \quad t_{2,n_2-1} \quad t_{2,n_2-1} \quad NaN \]

\[... \quad ... \quad ... \quad ... \quad ... \quad ...
\]

\[n_m \quad t_{m,1} \quad t_{m,2} \quad ... \quad t_{m,n_m-1} \quad t_{m,n_m-1} \quad ...
\]

where NaN is a non-numeric constant that does not participate in the procedure.

We evaluate the objective values of a population using (8). Observe that in the mathematical model described in the preceding section, the total inventory cost is a function of \((t_{j-1}, s_j, t_j)\) and \(s_j\) itself is a function of \((t_{j-1}, t_j)\) where the optimal \(s_j\) can be obtained from each pair of \((t_{j-1}, t_j)\) by solving (11). We employ the Stochastic Universal Sampling method (Baker, 1987) in the selection process and we assign the fitness of each individual using the Nonlinear Ranking method (Chipperfield et al., 1993).

We restrict the crossover to be performed between individuals of the same length only, similar to the multi-population GA whereby the information exchange is confined to those individuals within the same subpopulation only. We select the Discrete Recombination operator (Chipperfield et al., 1993) to perform the crossover.

We perform the process of mutation in two stages. In the first stage, we perform mutation on the integer \( n \) with a given probability where it is mutated to any integer greater than 2 but bounded by a fixed ceiling. Since the amount of replenishment points is dependent on \( n \), we devise a procedure whereby if \( n \) increases, then extra replenishment points, generated randomly, will be added accordingly. Then, we sort the new set of replenishment points in an ascending order. If \( n \) decreases, then we remove the last replenishment points from the set. This eliminates the need to sort the set again. Our limited experiments have shown that this procedure does not have a dramatic effect on the convergence of the algorithm. In the second stage, we perform mutation on the replenishment points using the Breeder GA (Mühlenbein and Schierkamp-Voosen, 1993). We mutate each ordering point with a probability of \(1/(n-2)\) where \((n-2)\) corresponds to the amount of replenishment points associated with each individual.

4. Numerical Examples

Our procedure was written using the Matlab high-level language and was run on a Pentium 4 IBM compatible machine running at 1.5 GHz in a Matlab 6.1 environment. Throughout all our experiments, the following values are used for the GA parameters (Chipperfield et al., 1993): number of individuals = 30, generation gap = 0.9, recombination rate = 0.9, mutation rate = 0.5, insertion rate = 0.9, selective pressure = 2, and maximal number of generations = 1000.

We illustrate our procedure using four example cases: two linear markets with the demand function \( D(t) = a + bt \) and two exponential markets with the demand function \( D(t) = a \exp(bt) \). The parameter values for these cases are shown in Table 1. For each case, we run our procedure 10 times and the results are shown in Table 2. Table 3 shows the optimal \( W \) from an optimal procedure (accurate to 5 decimal places) and the \( W^* \) from five heuristic procedures; these are described in the Appendices.
Table 1: Parameter values of the example cases

<table>
<thead>
<tr>
<th>Demand case</th>
<th>a</th>
<th>b</th>
<th>H</th>
<th>c₁</th>
<th>c₂</th>
<th>c₃</th>
<th>c₄</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear 1</td>
<td>6</td>
<td>1</td>
<td>11</td>
<td>9</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
<td>0.04</td>
</tr>
<tr>
<td>Linear 2</td>
<td>100</td>
<td>-10</td>
<td>4.78</td>
<td>1.5</td>
<td>0.2</td>
<td>0.8</td>
<td>0.4</td>
<td>0.04</td>
</tr>
<tr>
<td>Exponential 1</td>
<td>500</td>
<td>-0.98</td>
<td>11</td>
<td>9</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
<td>0.04</td>
</tr>
<tr>
<td>Exponential 2</td>
<td>0.1767</td>
<td>1</td>
<td>7</td>
<td>1.5</td>
<td>0.2</td>
<td>0.8</td>
<td>0.4</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 2: Results of the example cases using the GA procedure

<table>
<thead>
<tr>
<th>Demand case</th>
<th>Best $W$</th>
<th>Worst $W$</th>
<th>Average $W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear 1</td>
<td>55.04</td>
<td>55.04</td>
<td>55.04</td>
</tr>
<tr>
<td>Linear 2</td>
<td>31.13</td>
<td>31.34</td>
<td>31.20</td>
</tr>
<tr>
<td>Exponential 1</td>
<td>59.27</td>
<td>59.27</td>
<td>59.27</td>
</tr>
<tr>
<td>Exponential 2</td>
<td>21.79</td>
<td>22.49</td>
<td>21.93</td>
</tr>
</tbody>
</table>

Table 3: Results of the example cases using an optimal procedure and five heuristics

<table>
<thead>
<tr>
<th>Demand case</th>
<th>Linear 1</th>
<th>Linear 2</th>
<th>Exponential 1</th>
<th>Exponential 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>55.04</td>
<td>31.13</td>
<td>59.27</td>
<td>21.79</td>
</tr>
<tr>
<td>Heuristic 1</td>
<td>59.87</td>
<td>31.25</td>
<td>89.08</td>
<td>36.83</td>
</tr>
<tr>
<td>Heuristic 2</td>
<td>55.37</td>
<td>31.26</td>
<td>76.50</td>
<td>28.73</td>
</tr>
<tr>
<td>Heuristic 3</td>
<td>55.23</td>
<td>31.25</td>
<td>85.50</td>
<td>23.23</td>
</tr>
<tr>
<td>Heuristic 4</td>
<td>56.51</td>
<td>31.58</td>
<td>68.71</td>
<td>21.90</td>
</tr>
<tr>
<td>Heuristic 5</td>
<td>58.23</td>
<td>32.55</td>
<td>77.14</td>
<td>23.24</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, we found the near optimal solution for an inventory replenishment model with time-varying demand, constant deterioration and shortages using GA. Based on our numerical results, we have shown that our procedure performs better than some heuristical procedures. The best solutions found optimal values in all the data sets.
Appendix A: Optimal procedure

Referring to Section 2, the set of equations (10) gives us
\[ c_2 \int_{s_j}^{t_j} D(t)\,dt - c_a \exp(-\theta t) \int_{s_j}^{t_j} \exp(\theta t) D(t)\,dt = 0, \]
\[ j = 1, 2, ..., n - 2. \]
(13)

From this equation, observe that \( s_{j+1} > t_j \).

If we define \( t_0 = 0 \) and \( t_{n-1} = H \), it is easy to see that once \( s_1 = x \) is fixed, the sets of equations (11) and (13) will determine all other \( s_j \) and \( t_j \) recursively as functions of \( x \). To see this, let
\[ t_j = F(t_{j-1}, s_j), \]
(14)
\[ s_{j+1} = G(s_j, t_j). \]
(15)

where (14) and (15) are the solutions to equations (11) and (13) respectively. Now we have
\[ s_1 = x \]
\[ t_1 = F(0, x) \]
\[ s_2 = G(s_1, t_1) \]
\[ t_2 = F(t_1, s_2) \]
\[ \vdots \]
\[ s_{n-1} = G(s_{n-2}, t_{n-2}) \]
\[ t_{n-1} = F(t_{n-2}, s_{n-1}) \]

To summarize it, our procedure for finding the optimal replenishment policy for a given \( n \) is to fix \( s_1 = x \), obtain all the other \( t_j \) and \( s_j \) using (14) and solving (15) respectively, and finally check if \( t_{n-1} = H \). Since we have assumed \( n \) to be fixed, the optimal value of \( n \) can be obtained by repeating our procedure for different values of \( n \) \((n = 2, 3, \ldots)\) until the total inventory cost starts to increase.

We note that the second order conditions cannot be trivially checked, so we shall assume the sets of equations (9) and (11) are enough for determining the minimum of (8).

Appendix B: Heuristic procedures

The heuristic procedures used for comparison in Section 4 are based on those in Hariga and Benkherouf (1994). The variable \( n \) used in the following text will refer to the number of replenishments excluding the final back-order, since the final back-order only adds a constant value to the total inventory cost.

Heuristic 1: Constant Demand Approximation

In this heuristic, it assumed a constant demand rate over an infinite planning horizon. Let \( D \) be the average demand per unit of time, that is,
\[ D = \frac{1}{H} \int_0^H D(u)\,du. \]
(16)

For each replenishment cycle on the planning horizon, let \( T \) be the length of that cycle and \( I \) be the length of time during which inventory is carried \((I < T)\). Then, the amount of inventory carried during a cycle is
\[ I(0, I) = \int_0^T \frac{1}{\theta} (\exp(\theta t) - 1)\,D\,dt, \]
(17)
and the amount of shortage incurred during a cycle is
\[ S(I, T) = \int_I^T \int_0^T \frac{\theta}{D} \,D\,dt. \]
(18)

Finally, using (17) and (18), the total inventory costs per unit of time is
\[ TCUT(I, T) = \frac{1}{T} \left[ c_i + c_a I(0, I) + c_s S(I, T) \right]. \]

Then, the optimal \( I \) and \( T \) are the solutions of the following two equations:
\[ \frac{\partial TCUT}{\partial I} = F_1(I, T) = 0, \quad \text{and} \quad \frac{\partial TCUT}{\partial T} = G_1(I, T) = 0. \]
(19)
(20)

The number of replenishment orders placed during the planning horizon is then
\[ n = \left[ \frac{H}{T} \right], \]
where \([x]\) refers to the largest integer number which is smaller or equal to \( x \).
Next, we present the different steps of the heuristic method.

0. Compute \( c_a = c_2 + c_4 \theta \) and \( D \) from (16).
1. For \( I \in (0, H) \), compute \( T \) from (19) and use \((I,T)\) on (20).
2. If (20) is solved, go to 4.
3. If (20) is not solved, go to 1.
4. If \( T < H \), \( n = \lfloor H / T \rfloor \).
5. If \( T > H \), \( T = H \), and \( n = 1 \).
6. Find the replenishment schedule as follows:
   \[ t_j = jT \quad \text{for} \quad j = 0,1,\ldots,n-1 \quad \text{and} \quad t_a = H. \]
7. Compute the total inventory costs of the replenishment schedule.

Heuristic 2: Equal Replenishment Cycles

In this heuristic, it assumed that all replenishments are made at equal cycles of length \( T \). For \( n \) replenishments and \( j = 1,2,\ldots,n \), we have
\[ T_j = T, \quad t_j = jT, \quad s_j - t_{j-1} = I_j, \]
(11) gives us
\[ I_j + \frac{c_a}{c_5 \theta} [\exp(\theta I_j) - 1] - \frac{H}{n} = 0, \]
which has the solution \( I_j = I \) for all replenishment cycles. Moreover, observe that \( I = I(n) \).

Now, for the case of linear demand, (8) can be written as
\[
W(n) = n \frac{c_a}{\theta} \left[ \frac{a}{2} - \frac{b H (n-1)}{2 \theta} \right] + \left( \frac{H}{n} - 1 \right) \left[ \frac{a}{2} - \frac{b H (n+1) + 4}{4 \theta} \right],
\]
(22)
where \( \phi = \phi(n) = \exp(\theta I) - 1 \), and for the case of exponential demand as
\[
W(n) = n \frac{c_a}{\theta} \left[ \frac{\exp(\theta I) - 1}{\exp(\theta I) - 1} \right] + \left( \frac{H}{n} - 1 \right) \left[ \frac{\exp(\theta I) - 1}{\exp(\theta I) - 1} \right],
\]
(23)
where \( \psi = \psi(n) = (c_a \phi)/(c_5 \theta) \).
Assuming that the function \( W(n) \) is convex, the optimal number of orders under the restriction of equal replenishment cycles is the first integer which satisfies
\[ n W(n-1) < W(n) < W(n+1). \]

Next, we present the different steps of the heuristic method.

0. \( n = 1 \).
1. Compute \( W(n) \) and \( W(n+1) \).
2. If \( W(n) < W(n+1) \), stop, and go to 4.
3. If \( W(n) > W(n+1) \), \( n = n + 1 \), and go to 1.
4. Find the replenishment schedule as follows: \( t_j = jT \) for \( j = 1,2,\ldots,n \).
5. Compute the total inventory costs of the replenishment schedule.

Heuristic 3: Extended Silver-Meal Heuristic

In this heuristic, it determines each lot size sequentially, one at a time, by minimizing the total inventory costs per unit of time, rather than minimizing the total inventory costs up to the time horizon.

The total inventory costs per unit of time is
\[
TCUT(I,T) = \frac{1}{T} \left( c_1 + c_3 I(0,I) + c_3 S(I,T) \right),
\]
where
\[
I(0,I) = \int_0^I [\exp(\theta t) - 1] D(t) \, dt,
\]
(24)
and
\[
S(I,T) = \int_I^T D(u) \, du \, dt.
\]
(25)
The necessary conditions for the optimal $I$ and $T$ are

$$\frac{\partial \text{TCUT}}{\partial I} = F_2(I, T) = 0,$$

and

$$\frac{\partial \text{TCUT}}{\partial T} = G_3(I, T) = 0.$$  \hspace{1cm} (26) \hspace{1cm} (27)

Next, we present the different steps of the heuristic method.

0. Let $t_0 = 0$ and $j = 1$.

1. For $I_j \in (0, H)$, compute $T_j$ from (26) and use $(I_j, T_j)$ on (27).

2. If (27) is solved, go to 4.

3. If (27) is not solved, go to 1.

4. If $T_j = H$, go to 7.

5. If $T_j > H$, go to 8.

6. If $T_j < H$, let $t_j = t_{j-1} + T_j$, $a = a + bT_j$ (linear demand) or $a = a \exp(bT_j)$ (exponential demand), $H = H - T_j$, $j = j + 1$, and go to 1.

7. Let $t_j = t_{j-1} + T_j$, $n = j$, and go to 9.

8. Let $T_j = H$, $t_j = t_{j-1} + T_j$, $n = j$, and go to 9.

9. Compute the total inventory costs of the $(t_0, t_1, \ldots, t_n)$ schedule.

**Heuristic 4: The Extended Least Cost Heuristic**

In this heuristic, it equates the ordering cost with the sum of the holding, shortages, and deterioration costs to find each lot size sequentially and one at a time.

By such equalization, we have

$$c_1 - c_2 I(0, I) - c_3 S(I, T) = 0,$$

where $I(0, I)$ and $S(I, T)$ are from (24) and (25) respectively.

Next, we present the different steps of the heuristic method.

0. Let $t_0 = 0$ and $j = 1$.

1. For $I_j \in (0, H)$, compute $T_j$ from (27) and use $(I_j, T_j)$ on (28).

2. If (28) is solved, go to 4.

3. If (28) is not solved, go to 1.

Steps 4 to 9 are the same as in the Silver-Meal heuristic.

**Heuristic 5: The Extended Least-unit Cost Heuristic**

The procedure of this heuristic is the same as that of the Silver-Meal heuristic except for its cost function, which is the total cost per unit demand. The mathematical expression of this total cost per unit demand is given by

$$\text{TCUD}(I, T) = c_1 + c_2 I(0, I) + c_3 S(I, T) \int_0^T D(t) \, dt,$$

where $I(0, I)$ and $S(I, T)$ are from (24) and (25) respectively.

Then, the optimal $I$ and $T$ are the solutions of the following two equations:

$$\frac{\partial \text{TCUD}}{\partial I} = 0,$$

and

$$\frac{\partial \text{TCUD}}{\partial T} = 0.$$  \hspace{1cm} (30) \hspace{1cm} (31)

Next, we present the different steps in the heuristic method.

0. Let $t_0 = 0$ and $j = 1$.

1. For $I_j \in (0, H)$, compute $T_j$ from (30) and use $(I_j, T_j)$ on (31).

2. If (31) is solved, go to 4.

3. If (31) is not solved, go to 1.

Steps 4 to 9 are the same as in the Silver-Meal heuristic.
References


