OPTIMIZATION OF A MULTI-ITEM INVENTORY MODEL FOR
DETERIORATING ITEMS WITH CAPACITY CONSTRAINT
USING DYNAMIC PROGRAMMING

MAHDI KARIMI* AND SEYED JAFAR SADJADI
Department of Industrial Engineering, Iran University of Science and Technology
Tehran, Iran
(Communicated by Changjun Yu)

Abstract. In recent years, numerous studies have been conducted regarding
inventory control of deteriorating items. However, due to the complexity
of the solution methods, various real assumptions such as discrete variables and
capacity constraints were neglected. In this study, we presented a multi-item
inventory model for deteriorating items with limited carrier capacity. The
proposed research considered the carrier, which transports the order has limited
capacity and the quantity of orders cannot be infinite. Dynamic programming
is used for problem optimization. The results show that the proposed solution
method can solve the mixed-integer problem, and it can provide the global
optimum solution.

1. Introduction. The majority of the expenditures in every company is associ-
ated with inventory control. With proper management of the inventory costs, it is
possible to create a competitive advantage in the competitive market. Companies’
leadership can exploit this tool to make decisions (Wang et al.[19]). The economic
order quantity (EOQ) model has been used extensively over a century (Liao et
al.[8]). While studying inventory control, we should not ignore deterioration, which
is a deviation from expected performance (Sana et al.[13]). Some products will lose
their properties over time and will not be usable after a while. These products are
called deteriorating items. They include items such as fruits, meat, food, gasoline,
radioactive materials, photography films, high-tech products (Goyal and Giri[5]).
Therefore, considering an inventory model for deteriorating items can reduce the
cost of many related industries. In recent years, researchers have done many studies
in this field. Zhang et al.[22] studied a supply chain for deteriorating items with a
revenue sharing and cooperative investment contract and coordinated it. The main
idea of Tiwari et al. is investigating inflation and the impact of trade credit in a
two-warehouse environment for deteriorating items and find the retailer’s ordering
policy. Wu et al.[20] presented some inventory models for deteriorating items, and
they used innovative assumptions such as maximum lifetime under downstream par-
tial trade credits; they used discounted cash-flow analysis to credit-risk customers.
Teng et al.[17] studied inventory policies for deteriorating items with limited lifetime

* Corresponding author.

2020 Mathematics Subject Classification. Primary: 90B05, 90C26; Secondary: 90C39.
Key words and phrases. Inventory control, multi-item, deteriorating items, capacity constraint,
knapsack problem, dynamic programming.
and advanced payments. Samanta[12] developed a continuous production control inventory for deteriorating items, where shortages were allowed in this work. The work of Li et al.[7] investigated a joint pricing-replenishment-preservation technology investment problem for deteriorating items. Tiwari et al.[17] built an inventory model for deteriorating items for a supplier-retailer-buyer supply chain under two-level partial trade credit. Wu et al.[21] developed an inventory model for deteriorating items with expiration rates and optimized credit period.

Despite the extensive literature on this field, just a few studies consider multi-item models with capacity or budget constraints. Ghosh et al.[4] presented a multi-item inventory model for deteriorating items; they assumed that the store space is limited, and demand is stock-dependent. Capacity constraint and time-proportional backlogging rate are the contributions of the work of Dye et al.[3]. Chakraborty et al.[2] developed an inventory model for deteriorating items; they consider multi-warehouse, partial backlogging, and the effect of inflation to make their work more realistic. It should be noted that despite a long time since the first research in this field, this issue is still considered by many researchers so that many pieces of research are done annually in this field. Shah et al.[14] investigated quadratic demand and three echelon supply chain environment to integrate credit and ordering policies for deteriorating items. An economic order quantity for time-varying deteriorating items under two-level partial trade credit was presented by Mahata et al.[9]. Shah and Naik[15] presented inventory policies for deteriorating items and considered that the demand is time-price-backlog dependent. Multi-variate demand, inflation, and partial backlogging were the main assumptions of the deteriorating items inventory model presented by Malik and Sharma[10]. Karimi et al.[6] developed an economic order quantity model for deteriorating items; they considered that rework of the deteriorated products is allowed, and a fraction of these products can return to the inventory system. Rezagholifam et al.[11] studied an inventory model for non-instantaneous deteriorating items with capacity constraint and stock and price-dependent demand.

Inventory models for deteriorating items are mostly nonlinear models with complicated solution methods. Many realistic assumptions, such as discrete time variables, capacity constraints, and multi-item inventory systems, are ignored to simplify the solution method. Using a powerful tool to optimize these models, we can add realistic assumptions to develop more practical models. Dynamic programming is a systematic technique to determine the required information and how to use it properly to obtain more effective results (Bellman[1]). By using dynamic programming, we can consider new assumptions, such as the discrete time variables. By using this solution tool, it is possible to add the capacity or budget constraints by adding the upper and lower limits for the method, such as action and state, without changing the primary model.

A summary of the recent works is shown in Table 1.

In this paper, we present a multi-item inventory model for deteriorating items with time-varying demand and a consistent deterioration rate for each item. In this study, shortages with partial backlogging are allowed. There is a limit in the capacity of carrier, and we are not allowed to order an infinite quantity of products. Since in the real world, there is not any vehicle with unlimited capacity, this is a realistic assumption and makes the developed model more practical. The time horizon is finite, and the time is continuous, but the decision variables (the start time of shortage for each item) are discrete. The mentioned assumption is another
Table 1. A. Review of previous works

| Paper | Multi Item | Demand Function | Constraints          | Variables type | Solution method      | Shortages       |
|-------|------------|-----------------|----------------------|----------------|----------------------|-----------------|
| [7]   | No         | Constant        | Logical constraints  | Continuous     | Soft computing       | Allowed         |
| [2]   | Yes        | Stock-dependent | Capacity constraint  | Continuous     | Soft computing       | Allowed         |
| [8]   | No         | Constant        | No                   | Continuous     | Mathematical derivation | Not allowed    |
| [14]  | No         | Time-dependent  | Logical constraints  | Continuous     | Soft computing       | Not allowed    |
| [9]   | No         | Trade credit-dependent | No                   | Continuous     | Soft computing       | Not allowed    |
| [15]  | No         | Time-price backlog dependent | No                   | Continuous     | Mathematical derivation | Allowed         |
| [6]   | No         | Time-dependent  | No                   | Continuous     | Mathematical derivation | Allowed         |
| [11]  | No         | Stock and price dependent | Capacity constraint  | Continuous     | Mathematical derivation | Not allowed    |
| This Paper | Yes       | Time-dependent  | Capacity constraint  | Discrete and continuous | Dynamic Programming | Allowed         |

realistic assumption, which makes ordering planning easier (for example, it is easier to enter the shortage in the first hour of a working day rather than the midnight of a holiday). The purpose of the model is to determine when each item enters shortage (if we know the shortage start time, we can calculate the ordering quantity). The considered problem is optimized using dynamic programming. At last, a numerical example is provided to illustrate the implementation of the proposed model. The graphical representation of the inventory model for different demand functions is shown in Figure 1 (This example includes four products, but the proposed model is general and can include an arbitrary amount of products).

2. Highlights and contribution of this paper.

- This paper presented a multi-item inventory model.
- Unlike previous works that had to use continuous variables in order to be able to use the derivation method, in this paper, discrete variables were used to make the proposed model more realistic.
- Dynamic programming was used to solve the proposed model, which includes constraint and discrete variables.
- Realistic constraint (capacity) was added to the model.
- The solution method is such that small examples can be solved manually, and there is no need to use software computing.

3. Methodology. In this section, the following notations and assumptions are used to build the model, and the developed mathematical model will be presented.
3.1. **Notations.** The following notations are used in the presented model.

**Indices**

- \( p \): indicator to show the product

**Discrete variable (Independent decision variable)**

- \( t_{s_p} \): shortage start time for the \( p \)th product

**Continuous variables**

- \( t \): time variable
- \( Q_p \): order quantity of the \( p \)th product
- \( Q'_p \): the quantity of the \( p \)th product that is needed to fulfill the backlogged demand
- \( HI_p \): inventory held of the \( p \)th product
- \( SI_p \): shortage amount of the \( p \)th product
- \( BI_p \): lost sale of the \( p \)th product
- \( PI_p \): the quantity of the \( p \)th product that is needed to fulfill both satisfied and backlogged demand
- \( I_p(t) \): inventory amount of the \( p \)th product
- \( THC_p \): total holding costs for the \( p \)th product over the time horizon
- \( TSC_p \): total shortage costs for the \( p \)th product over the time horizon
- \( TBC_p \): total credit lost costs for the \( p \)th product over the time horizon
- \( TPC_p \): total purchase costs for the \( p \)th product over the time horizon
- \( TC \): total costs of inventory model

**Parameters**

- \( \vartheta_p \): deterioration rate for \( p \)th product
- \( \delta_p \): backlog rate for the \( p \)th product
- \( v_p \): volume of the \( p \)th product
- \( D_p(t) \): demand function of the \( p \)th product in time \( t \)
- \( HC_p \): holding cost for the \( p \)th product per unit, per time
- \( SC_p \): shortage cost for the \( p \)th product per unit, per time
- \( BC_p \): credit lost cost for the \( p \)th product per unit, per time

**Figure 1. Inventory level of each item vs. time.**
3.2. Assumptions. The proposed model was built based on the following assumptions.

- The deterioration rates of products are different and constant over the time horizon.
- The demand rate of each product ($D_p(t)$) is a continuous, non-negative, monotonic function of time.
- Shortage is allowed for each product, and the inventory level at the end of horizon time could be positive, zero, or negative.
- When the shortage occurs, the backlogging rate depends on time, and the numbers of customers who wait for the next replenishment are decreasing. So, the backlogging rate in time $t$ is calculated by $1/(1 + \delta_p(PH - t))$ for $ts_p \leq t \leq PH$ and $\delta_p > 0$.
- Replenishment time is at the beginning of the time horizon ($t = 0$), and the lead time is negligible.
- Carrier space is limited, and the total volume of ordered products should be less than $Cap$.
- All of the unit costs ($HC_p, SC_p, BC_p$, and $PC_p$) remain constant over the time horizon.
- The time horizon is continuous, but discrete time variables are used to make the proposed model more realistic.
- It is assumed that items are independent, and the ordering quantity of each product has no effect on another.

3.3. Mathematical model. In this section, we develop a multi-item inventory model for deteriorating items considering ordering capacity constraints. In the proposed model, shortage is allowed with partial backlogging, and the demand rate of each product is time-related. The main idea of this study is to add realistic constraints such as capacity and budget constraints in multi-item models and propose a proper solution to solve these models. We first calculate each inventory cost separately, then the final model will be presented.

The inventory amount of the $p$th item ($I_p(t)$) at any time $t$ over the holding period ($0 \leq t \leq ts_p$) is given by the following equation:

$$\frac{dI_p(t)}{dt} = \vartheta_p I_p(t) - D_p(t), \text{ for } 0 \leq t \leq ts_p,$$  \hspace{1cm} (1)

We can calculate $I_p(t)$ by solving equation 1, then we have:

$$I_p(t) = e^{-\vartheta_p t} \int_t^{ts_p} e^{\vartheta_p t'} D_p(t') dt', \text{ for } 0 \leq t \leq ts_p,$$  \hspace{1cm} (2)

With boundary conditions, the ordering quantity is:

$$Q_p = I_p(0) = e^{-\vartheta_p 0} \int_0^{ts_p} e^{\vartheta_p t'} D_p(t') dt' = \int_0^{ts_p} e^{\vartheta_p t'} D_p(t') dt,$$  \hspace{1cm} (3)

The amount of the $p$th item held is written as below:
\[ HI_p = \int_0^{t_{sp}} I_p(t) dt = \frac{1}{\vartheta_p} \int_0^{t_{sp}} (e^{\vartheta_p t} - 1)D_p(t) dt, \quad (4) \]

So, the total holding costs for \( p \)-th item can be obtained from the following equation:

\[ THI_p = HC_p HI_p. \quad (5) \]

Over the shortage period \((t_{sp} \leq t \leq PH)\), change in the inventory level is showed as the following equation:

\[ \frac{dI_p(t)}{dt} = -\frac{D_p(t)}{1 + \delta_p(PH - t)}, \quad \text{for} \quad t_{sp} \leq t \leq PH, \quad (6) \]

By solving equation 6, the amount of inventory of \( p \)-th item, in time \( t \) in shortage period is obtained as below:

\[ I_p(t) = -\int_{t_{sp}}^{t} \frac{D_p(t)}{1 + \delta_p(PH - t)}, \quad \text{for} \quad t_{sp} \leq t \leq PH, \quad (7) \]

From equation 7, we get the shortage amount for each product as the following equations:

\[ SI_p = \int_{t_{sp}}^{t} |I_p(t)| dt = \int_{t_{sp}}^{PH} \frac{D_p(t)(PH - t)}{1 + \delta_p(PH - t)} dt, \quad (8) \]

Note that, from equation 7, we know for any given time \( t \) the inventory level is negative. In equation 8, we want to calculate the shortage amount of each item, which should be a positive value. Hence, the absolute value of \( I_p(t) \) is used in equation 8. From equation 8, the amount of each product that is needed to fulfill the backlogged demand can be obtained, the number of customers who wait for the next replenishment is \( SI_p \); hence we have:

\[ Q_p' = SI_p = \int_{t_{sp}}^{PH} \frac{D_p(t)(PH - t)}{1 + \delta_p(PH - t)} dt, \quad (9) \]

From equation 3, and equation 9, we get the total amount of the \( p \)-th product that is needed to fulfill both satisfied and backlogged demand as the following equation:

\[ PI_p = Q_p + Q_p' = \int_0^{t_{sp}} e^{\vartheta_p t}D_p(t) dt + \int_{t_{sp}}^{PH} \frac{D_p(t)(PH - t)}{1 + \delta_p(PH - t)} dt, \quad (10) \]

Total shortage costs and total purchase costs for each item is given by:

\[ TSI_p = SC_p SI_p, \quad (11) \]

\[ TPI_p = PC_p PI_p. \quad (12) \]

The backlogging fraction \((1/(1 + \delta_p(PH - t)))\) is used to calculate the number of customers who wait for the next replenishment, hence the fraction of customers who are not waiting to satisfy their demand is \( 1 - 1/(1 + \delta_p(PH - t)) \), so we have:

\[ BI_p = \int_{t_{sp}}^{PH} (1 - \frac{1}{1 + \delta_p(PH - t)})D_p(t) dt = \delta_p \int_{t_{sp}}^{PH} \frac{D_p(t)(PH - t)}{1 + \delta_p(PH - t)} dt, \quad (13) \]

we have:

\[ TBI_p = BC_p BI_p. \quad (14) \]

At last, the total costs of the inventory system are:

\[ TC = \sum_p (THI_p + TSI_p + TPI_p + TBI_p). \quad (15) \]
Since the capacity of the carrier vehicle is not unlimited, the total volume of
ordered items could be calculated from equation 3, so we have the following con-
straint:

$$\sum_p Q_p v_p \leq Cap,$$

(16)

Therefore, the problem is written as below:

Minimize $$TC\{t_{sp}\} = \sum_p (THI_p + TSI_p + TPI_p + TBI_p)$$

Subject to:

$$\sum_p Q_p v_p \leq Cap,$$

(17)

We can also write the extended form of equation 17 as the following model:

Minimize $$TC\{t_{sp}\} = \sum_p \left( \frac{HC_p}{\vartheta_p} \int_0^{t_{sp}} (e^{\vartheta_p t} - 1) D_p(t) dt \right)$$

$$+ (SC_p + PC_p + \delta_p BC_p) \int_{t_{sp}}^{PH} \frac{D_p(t)(PH - t)}{1 + \delta_p(PH - t)} dt + PC_p \int_0^{t_{sp}} e^{\vartheta_p t} D_p(t) dt$$

Subject to:

$$\sum_p (v_p \int_0^{t_{sp}} e^{\vartheta_p t} D_p(t) dt) \leq Cap.$$

(18)

In this section, the proposed model is presented; in the next section, dynamic
programming is implemented to solve the problem.

3.4. Optimization. The most practical method to solve inventory models for de-
teriorating items is the derivative method. Since the proposed model has discrete
decision variables, it is impossible to solve it using this method. On the other hand,
the proposed model is nonlinear and has a capacity constraint; the usual meth-
ods that can solve these models are complicated and are not easy to use. If we
want to solve the problem using common methods, it is necessary to prove that the
whole model (both objective function and constraint) is convex/concave to show
the obtained solution is the global minimum/maximum. Forasmuch as the primary
decision variables are discrete, we must use graphical charts to show the convexity,
but the visual proof is not accurate enough to be used in general.

In this study, we used dynamic programming to overcome these problems. This
method can solve a wide range of problems, such as discrete problems, in a simple
way. It can be used in both linear and nonlinear problems. Since this method
searches all of the feasible points, it is not necessary to show the model convexity
of the model, and the solution is always the global optimum solution. Unlike other
methods, in this method, it is easy to add constraints to the model by determining
the upper and lower limits for some of the characteristics of the method (such as
state and action). In the next subsection, we present the solution method.

Proposed backward dynamic programming In this section, the proposed
solution approach is presented. First, we explain a few terms and notations used in
this method.

- It is necessary to break the problem into several sub-problems to use dynamic
programming. Each of these sub-problems is called a “stage” and is denoted
In this study, in each stage, we allocate some of the remaining capacity to order one specific item, it means each sub-problem is allocating some of the remaining capacity to order each product regardless of the other products. In fact, we use nonlinear programming to solve a single-item inventory model in each sub-problem. Note that the number of stages is as large as the number of items (for each \( p \), we have one \( n \), so in this section, in all of the model notations, we use \( n \) instead of \( p \)). It is essential to know the priority of choosing products for each stage does not matter, and the final results are the same; it means if we assign item 1 to stage 1 and item 2 to stage 2, the results will be the same as when we set item 1 to stage 2 and item 2 to stage 1.

In stage \( n \), the ordering quantity of the \( n \)th item is called the “action” and is showed by \( k_n \), where the optimum action of the \( n \)th stage is \( k^*_n \). In this method, \( k_n \) is \( Q_n \) and must has a continuous value, but it is easier to work with discrete values as the actions. Because it is necessary to use the derivative to determine the optimum value of continuous variables and for a nonlinear model, such as equation 18, it is hard (or even impossible) to use derivative. On the other hand, we can check all discrete values to obtain the optimum value (if the number of discrete variables is not very large). Regarding equation 3, \( Q_n \) is related to \( ts_n \), and if we know \( ts_n \) we can calculate \( Q_n \). In fact, \( Q_n \) is continuous, but according to equation 3 it cannot take any arbitrary value, and the number of different values of \( Q_n \) is limited by \( ts_n \). So we use \( ts_n \) as the action and show it by \( k'_n \), this lets us work with discrete actions instead of continuous ones. The relation between \( k_n \) and \( k'_n \) is:

\[
k_n = \int_0^{k'_n} e^{\vartheta_n t} D_n(t) dt.
\]

Then, for a specific stage \( n \), we can rewrite equation 17 as the following model:

Minimize \((THI_n + TSI_n + TPI_n + TBI_n)\)

Subject to:

\[
k_n v_n \leq \text{Cap} - \sum_{n'=1}^{n-1} k_{n'} v_{n'}.
\]

Note that equation 17, equation 18 and, equation 20 are the same equations with different notations.

In this method, the “state” is the amount of resource we have at the beginning of each stage and is denoted by \( i_n \). Here, the state is the amount of the capacity we have at the beginning of the stage. Hence the state has continuous values but similar to what we argued in the previous part; we can use the discrete time variable instead of the continuous quantity variable and show it by \( i'_n \). Suppose that the remaining capacity at the beginning of the stage is equal to \( i_n \); since the fraction of \( i_n \) that we can allocate to the \( n \)th product is related to \( ts_n \), we can calculate the state value by the following equation:

\[
i'_n = \{ts_n \int_0^{ts_n} e^{\vartheta_n t} D_n(t) dt \leq i_n \}.
\]

we have \( ts_n = k'_n \) then,

\[
i'_n = \{k'_n \int_0^{k'_n} e^{\vartheta_n t} D_n(t) dt \leq i_n \}.
\]

Note that equation 17, equation 18 and, equation 20 are the same equations with different notations.
Since each stage is related to the previous stage, we need to pay more attention to determine the amount of \( i_n \) in the higher stages. It is necessary to consider the combinations of different values of \( i_n \) in this stage and the preceding stages. In the higher stages, the number of possible values for \( i_n \) increases, but the number of these states is finite and can be considered as a discrete value.

- The amount of income(cost) earned in the stage \( n \) with the state \( i_n \) under the action \( k'_n \) is called the “revenue” and is showed by \( r(n, i_n, k'_n) \). According to equation 20, the revenue can be calculated using the formula below.

\[
r(n, i_n, k'_n) = THI_n + TSI_n + TPI_n + TBI_n.
\]  

- In dynamic programming, it is assumed that each stage is only associated with the stage before and after it. This relationship is shown using the “transition function”. In the proposed method, the remaining capacity at the end of each stage is showed by \( j_n \) and is yielded using the transition function as below:

\[
j_n = i_n - k_nv_n, \quad \text{and} \quad j_n = i_{n-1}. \tag{23}
\]

Note that we always have \( 0 \leq j_n \) to ensure that the capacity constraint is not violated.

- The basis of computation in dynamic programming is the “recursive function” and is denoted by \( f(n, i_n) \). This function is an optimization model that determines the optimum action of each state, which showed by \( k^*_n \). The recursive function for the proposed method is:

\[
f(n, i_n) = \text{Minimize} \{ r(n, i_n, k'_n) + f(n - 1, j_n) \}. \tag{24}
\]

In the first stage, we have \( f(n - 1, j_n) = 0 \).

In this study, the developed model is similar to the knapsack problem and the presented method is based on backward dynamic programming. In the proposed method in the first stage, it is assumed that the allocation of all stages is done, and only the allocation of the last product remains. To calculate the state of this stage, we need to know how much capacity is required in the next stages, then put the amount of remaining capacity as the state of the current stage. In the next stage, we assume that the allocation of all stages, except for the last two steps, has taken place and calculate the state as said before. This process continues until the final stage. In the final stage, since we allocate capacity to the first product, we have all the available capacity, so in the last stage \( i_n = Cap \). In general, we must investigate the various values of the state that may exist at each stage. In this method, the allocation of the last product is carried out in the first stage; because of this reverse trend, this method is called the backward method (in the forward method, we investigate the first item in the first stage and continue this method until the last item). To illustrate the method, a flowchart is provided and shown in Figure 2. In this section, the solution method is presented. We illustrate the model and the solution method by a numerical example in the next section.

4. **Numerical example.** In this section, a numerical example is provided to illustrate the mathematical model and the optimization method. The following parameters are used in this example:

**Example parameters**

\( p = 3 \) (number of products)
\[ \vartheta_p = (0.02, 0.015, 0.03) \]
\[ \delta_p = (12, 25, 16) \]
\[ v_p = (1.5, 2, 1) \]
\[ D_p(t) = (110 - t^2, 35 + e^t, 150 + t^4) \]
\[ HC_p = (0.1, 0.15, 0.1) \]
\[ SC_p = (12, 25, 8.5) \]
\[ BC_p = (25, 36, 28) \]
\[ PC_p = (18, 20, 21) \]
\[ PH = 5 \]
\( Cap = 800 \)

**Solution**

There are three stages (one for each item) in this example. We obtain the global optimum solution using the proposed method.

**Stage 1: \( n = 1 \), the first stage, the first item**

In this stage we have \( PH = 5 \), and replenishment (ordering) time is zero, so \( k_1' \) can be six different values (0 to 5). The required space for each action is shown in Table 2.

**Table 2.** The required and remaining space for each action in the stage 1

| \( k_1' \) | 0   | 1   | 2   | 3   | 4   | 5   |
|-----------|-----|-----|-----|-----|-----|-----|
| \( k_1 \) | 0   | 110.7 | 221.7 | 330.7 | 435.4 | 533.5 |
| \( k_1v_1 \) | 0   | 166.05 | 332.55 | 496.05 | 653.1 | 800.25 |
| \( j_1 \) | 800  | 633.95 | 467.45 | 303.95 | 146.9 | -0.25 |

According to this table \( k_1' = 5 \) is infeasible because \( Cap < 800.25(j_1 < 0) \), so the vehicle cannot carry this amount of the first product.

In the first stage, the state can be a continuous value between 0 and 800 \( (Cap) \), but we do not know its exact value, so we should consider different values of it.

Different values of state are shown in Table 3.

**Table 3.** Different values of the state in the stage 1

| \( i_1 \) | 0 \( \leq i_1 < 166.05 \) | 166.05 \( \leq i_1 < 332.55 \) | 332.55 \( \leq i_1 < 496.05 \) | 496.05 \( \leq i_1 < 653.1 \) | 653.1 \( \leq i_1 \) |
|-----------|-----------------|-----------------|-----------------|-----------------|-----------------|
| \( i_1' \) | \{0\}           | \{0.1\}         | \{0.1,2\}       | \{0,1,2,3\}     | \{0,1,2,3,4\}   |

Different values of the revenue are calculated using equation 22.

\[
\begin{align*}
  r(1, 0 \leq i_1 < 166.05, 0) &= 13099; \\
  r(1, 166.05 \leq i_1 < 332.55, 0) &= 13099; \\
  r(1, 332.55 \leq i_1 < 496.05, 1) &= 12686; \\
  r(1, 496.05 \leq i_1 < 653.1, 0) &= 13099; \\
  r(1, 653.1 \leq i_1 \leq 800, 0) &= 13099; \\
  r(1, 653.1 \leq i_1 \leq 800, 1) &= 12868; \\
  r(1, 653.1 \leq i_1 \leq 800, 2) &= 12346; \\
  r(1, 653.1 \leq i_1 \leq 800, 3) &= 12095; \\
  r(1, 653.1 \leq i_1 \leq 800, 4) &= 11964.
\end{align*}
\]

Now we should determine \( f(n, i_n) \) according to equation 24.

\[
\begin{align*}
  f(1, 0 \leq i_1 < 166.05) &= \text{minimize } \{ r(1, 0 \leq i_1 < 166.05, k_1') + 0 \} = 13099 \text{ for } k_1' = 0; \\
  f(1, 166.05 \leq i_1 < 332.55) &= 12686 \text{ for } k_1' = 1; \\
  f(1, 332.55 \leq i_1 < 496.05) &= 12346 \text{ for } k_1' = 2; \\
  f(1, 496.05 \leq i_1 < 653.1) &= 12095 \text{ for } k_1' = 3; \\
  f(1, 653.1 \leq i_1 \leq 800) &= 11964.
\end{align*}
\]
\( f(1.653.1 \leq i_1 \leq 800) = 11964 \) for \( k_1^* = 4 \).

**Stage 2:** \( n = 2 \), the second stage, the second item.

\( k_2 \) can be six values, so the required and remaining space for each action is shown in Table 4.

**Table 4. The required space for each action in the stage 2**

| \( k_2' \) | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|---|---|---|---|---|---|
| \( k_2 \) | 0 | 36.9 | 77.5 | 127.1 | 200.4 | 338.3 |
| \( k_2v_1 \) | 0 | 73.8 | 155 | 254.2 | 400.8 | 676.6 |
| \( j_2 \) | 800 | 726.2 | 645 | 545.8 | 399.8 | 123.4 |

In this stage, we should combine different values of \( i_1 \) and \( k_2v_2 \) to obtain other values of \( i_2 \). Different values of \( i_2 \) and \( i_2' \) shown in Table 5.

**Table 5. Different values of the state in the stage n=2**

| \( i_2 \) | \( 0 \leq i_2 \) | \( 73.8 \leq i_2 \) | \( 155 \leq i_2 \) | \( 166.05 \leq i_2 \) | \( 240.3 \leq i_2 \) |
|-----------|-----------------|-----------------|-----------------|-----------------|-----------------|
| \( i_2' \) | \( < 73.8 \) | \( < 155 \) | \( < 166.05 \) | \( < 240.3 \) | \( < 254.2 \) |
| \( i_2 \) | \{0\} | \{0.1\} | \{0.1,2\} | \{0.1,2,3\} | \{0.1,2,3,4\} |
| \( i_2' \) | \{1\} | \{0,1\} | \{0,1,2\} | \{0,1,2,3\} | \{0,1,2,3,4\} |
| \( i_2 \) | \{0,1\} | \{0,1,2\} | \{0,1,2,3\} | \{0,1,2,3,4\} | \{0,1,2,3,4,5\} |
| \( i_2' \) | \{0,1,2\} | \{0,1,2,3\} | \{0,1,2,3,4\} | \{0,1,2,3,4,5\} | \{0,1,2,3,4,5,6\} |

We can obtain \( r(2, i_2, k_2) \) using equation 22. For any value of \( i_2 \) we have:

\( r(2, i_2, 0) = 11305; \quad r(2, i_2, 1) = 11163; \quad r(2, i_2, 2) = 11033; \)

\( r(2, i_2, 3) = 10910; \quad r(2, i_2, 4) = 10799; \quad r(2, i_2, 5) = 11278. \)

\( f(2, i_2) \) is calculated with the same process as the previous stage; these values are shown in Table 6.

**Stage 3:** \( n = 3 \), the final stage, the third item.

Required and remaining space for each action in this stage is shown in Table 7.

Given what was said in the explanation of the method, in the last stage, the state is equal to all of the available capacity, so \( i_3 = \text{Cap} \).

We can calculate \( r(3, i_3, k_3) \) and \( j_3 \) as below:

\( r(3, i_3, 0) = 36914 \) and \( j_3 = 800; \)
Table 6. The recursive function in the second stage

| $i_2$ | $0 \leq i_2 < 73.8$ | $73.8 \leq i_2 < 155$ | $155 \leq i_2 < 166.05$ | $166.05 \leq i_2 < 240.3$ | $240.3 \leq i_2 < 254.2$ |
|-------|------------------|-------------------|-------------------|-------------------|-------------------|
| $f(2, i_2)$ | $k_1^* \begin{cases} k_1^* \\ 0 \end{cases}$ | $k_2^* \begin{cases} k_2^* \\ 0 \end{cases}$ | $k_1^* \begin{cases} k_1^* \\ 0 \end{cases}$ | $k_2^* \begin{cases} k_2^* \\ 2 \end{cases}$ | $k_1^* \begin{cases} k_1^* \\ 0 \end{cases}$ |

| $i_2$ | $254.2 \leq i_2 < 321.5$ | $321.5 \leq i_2 < 332.55$ | $332.55 \leq i_2 < 400.8$ | $400.8 \leq i_2 < 406.3$ | $406.3 \leq i_2 < 420.7$ |
|-------|------------------|-------------------|-------------------|-------------------|-------------------|
| $f(2, i_2)$ | $23849 \begin{cases} k_1^* \\ \frac{1}{2} \end{cases}$ | $23849 \begin{cases} k_1^* \\ \frac{1}{2} \end{cases}$ | $23849 \begin{cases} k_1^* \\ \frac{1}{2} \end{cases}$ | $23849 \begin{cases} k_1^* \\ \frac{1}{2} \end{cases}$ | $23849 \begin{cases} k_1^* \\ \frac{1}{2} \end{cases}$ |

| $i_2$ | $420.7 \leq i_2 < 487.5$ | $487.5 \leq i_2 < 496.05$ | $496.05 \leq i_2 < 567.3$ | $567.3 \leq i_2 < 596.7$ | $596.7 \leq i_2 < 586.7$ |
|-------|------------------|-------------------|-------------------|-------------------|-------------------|
| $f(2, i_2)$ | $23509 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23509 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23509 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23509 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23509 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ |

| $i_2$ | $586.7 \leq i_2 < 650.9$ | $650.9 \leq i_2 < 676.6$ | $676.6 \leq i_2 < 726.9$ | $726.9 \leq i_2 < 733.3$ |
|-------|------------------|-------------------|-------------------|-------------------|
| $f(2, i_2)$ | $23256 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23256 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23256 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23256 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ |

| $i_2$ | $733.3 \leq i_2 < 750.1$ | $750.1 \leq i_2 < 800$ |
|-------|------------------|-------------------|
| $f(2, i_2)$ | $23127 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ | $23127 \begin{cases} k_1^* \\ \frac{2}{2} \end{cases}$ |

Table 7. The required and remaining space for each action in stage $n=3$

| $k_3$ | 0 | 1 | 2 | 3 | 4 | 5 |
|-------|---|---|---|---|---|---|
| $k_3$ | 0 | 152.4 | 315.9 | 523.2 | 863.8 | 1517.5 |
| $k_3v_3$ | 0 | 152.4 | 315.9 | 523.2 | 863.8 | 1517.5 |
| $j_3$ | 800 | 647.6 | 484.1 | 276.8 | -63.8 | -717.5 |

$r(3, i_3, 1) = 36204$ and $j_3 = 647.6$;
$r(3, i_3, 2) = 35601$ and $j_3 = 484.1$;
$r(3, i_3, 3) = 35054$ and $j_3 = 276.8$.

Regarding equation 24, $j_3$, and Table 5, $f(3, i_3 = Cap)$ can be obtained.
Hence: $f(3, i_3 = Cap) = 58903$ for $k_1^* = 1, k_2^* = 1, k_3^* = 3$.
We have $f(3, i_3 = Cap) = TC\{ts_p\}, k_n^* = ts_p$ and, $k_n^* = Q_p$.

Therefore, the optimum solution is: $ts_1 = 1, ts_2 = 1, ts_3 = 3, Q_1 = 110.7, Q_2 = 36.9, Q_3 = 523.2$
and, $TC\{ts_p\} = 58903$. 


The remaining capacity is 36.5, and the capacity constraint is not violated, so the optimum solution is feasible. Note that, the obtained solution is the global minimum of the inventory system.

The inventory level of each item over time is presented in Figure 3. The result shows that since the demand function of the third item is exponential, the maximum amount of vehicle capacity is assigned to the purchase of this item. During the shortage period, the demand for product 3 increases dramatically over time. Given the high costs of shortages and lost sales, if the amount of shortage of this product increases, the total costs will increase. So, in order to reduce the costs of shortages, the order quantity of this item is more than the other items. Notice that, according to the capacity constraint, $t_{s3} = 3$ is the maximum possible value for this variable.

In this section, a numerical example is solved in detail to illustrate the solution method.

![Figure 3. The inventory level of each item over time](image)

5. Managerial insights.

- According to the results, it is necessary to pay attention to the demand function of each product; products with exponential demand function will increase the shortage and opportunity lost costs and we must increase the ordering quantity of these products, and we lost the opportunity to order other items.
- The manager can focus on items with linear demand rates. If it is necessary to order items with exponential demand function, the company must make a trade-off between the costs and benefits of these items.
- This paper showed that we could add realistic assumptions (such as ordering in discrete times) to the model, and the obtained results can be used in the real world. The manager should use these assumptions to make the results more usable and reliable.
• The results of this paper showed that the storages play an important role in inventory systems, and we cannot ignore them easily (like many of previous works). The manager should pay attention to the shortages to make customers happy and do not let them leave the company easily.

6. Conclusions. Many researchers studied inventory models for deteriorating items, but just a few studies consider multi-item models with capacity or budget constraints. This is because of the nonlinear behavior of these models and the complexity of nonlinear optimization methods. On the other hand, because of simplification of solution methods, representing non-convex models (like models with allowable quantity discounts), which are commonly being used, has been avoided. Hence, representing an optimization method, which can fill a part of the mentioned gap, is drastically significant. In this study, we presented a multi-item inventory model for deteriorating items with allowable shortage. Discrete time variables and capacity constraint are used to make the proposed model more realistic. Due to the discrete variables and capacity constraint, the presented model cannot be optimized by conventional methods. Dynamic programming is implemented to solve the problem. By providing a numerical example, we showed that the proposed solution method can solve these problems. The results show that most of the available capacity is allocated to items with a higher amount of shortages to reduce the shortage costs.

For further studies, the proposed model could have several extensions. We can extend the model by considering more realistic assumptions such as uncertainty in demand rate. As well as, investigating the effect of discount, inflation, and advanced payments could be interesting.

Acknowledgments. The funding sources were not provided any financial support to conduct this study. The authors would like to thank the editor-in-chief, the associated editor, and the anonymous reviewers for their constructive comments on earlier versions of this paper.

REFERENCES

[1] R. Bellman, Dynamic Programming, *Science*, **153** (1966), 34–37.
[2] D. Chakraborty, D. K. Jana and T. K. Roy, Multi-warehouse partial backlogging inventory system with inflation for non-instantaneous deteriorating multi-item under imprecise environment, *Soft Computing*, **24** (2020), 14471–14490.
[3] C.-Y. Dye, L.-Y. Ouyang and T.-P. Hsieh, Deterministic inventory model for deteriorating items with capacity constraint and time-proportional backlogging rate, *European J. Oper. Res.*, **178** (2007), 789–807.
[4] S. K. Ghosh, T. Sarkar and K. Chaudhuri, A multi-item inventory model for deteriorating items in limited storage space with stock-dependent demand, *American Journal of Mathematical and Management Sciences*, **34** (2015), 147–161.
[5] S. K. Goyal and B. C. Giri, Recent trends in modeling of deteriorating inventory, *European J. Oper. Res.*, **134** (2001), 1–16.
[6] M. Karimi, S. J. Sadjadi and A. G. Bijaghini, An economic order quantity for deteriorating items with allowable rework of deteriorated products, *J. Ind. Manag. Optim.*, **15** (2019), 1857–1879.
[7] G. Li, X. He, J. Zhou and H. Wu, Pricing, replenishment and preservation technology investment decisions for non-instantaneous deteriorating items, *Omega*, **84** (2019), 114–126.
[8] J.-J. Liao, K.-N. Huang, K.-J. Chung, S.-D. Lin, S.-T. Chuang and H. M. Srivastava, Optimal ordering policy in an economic order quantity (EOQ) model for non-instantaneous deteriorating items with defective quality and permissible delay in payments, *Rev. R. Acad. Cien. Exactas Fis. Nat. Ser. A Mat. RACSAM*, **114** (2020), Paper No. 41, 26 pp.
[9] P. Mahata, G. C. Mahata and S. K. De, An economic order quantity model under two-level partial trade credit for time varying deteriorating items, *International Journal of Systems Science: Operations and Logistics*, 7 (2020), 1–17.

[10] A. K. Malik and A. Sharma, An inventory model for deteriorating items with multi-variate demand and partial backlogging under inflation, *International Journal of Mathematical Sciences*, 10 (2011), 315–321.

[11] M. Rezagholifam, S. J. Sadjadi, M. Heydari and M. Karimi, Optimal pricing and ordering strategy for non-instantaneous deteriorating items with price and stock sensitive demand and capacity constraint, *International Journal of Systems Science: Operations and Logistics*, (2020).

[12] G. P. Samanta and A. Roy, A production inventory model with deteriorating items and shortages, *Yugosl. J. Oper. Res.*, 14 (2004), 219–230.

[13] S. Sana, S. K. Goyal and K. S. Chaudhuri, A production-inventory model for a deteriorating item with trended demand and shortages, *European J. Oper. Res.*, 157 (2004), 357–371.

[14] N. H. Shah, U. Chaudhari and L. E. Cárdenas-Barrón, Integrating credit and replenishment policies for deteriorating items under quadratic demand in a three echelon supply chain, *International Journal of Systems Science: Operations and Logistics*, 7 (2020), 34–45.

[15] N. H. Shah and M. K. Naik, Inventory policies for deteriorating items with time-price backlog dependent demand, *International Journal of Systems Science: Operations and Logistics*, 7 (2020), 76–89.

[16] J.-T. Teng, L. E. Cárdenas-Barrón, H.-J. Chang, J. Wu and Y. Hu, Inventory lot-size policies for deteriorating items with expiration dates and advance payments, *Appl. Math. Model.*, 40 (2016), 8605–8616.

[17] S. Tiwari, L. E. Cárdenas-Barrón, M. Goh and A. A. Shaikh, Joint pricing and inventory model for deteriorating items with expiration dates and partial backlogging under two-level partial trade credits in supply chain, *International Journal of Production Economics*, 200 (2018), 16–36.

[18] S. Tiwari, L. E. Cárdenas-Barrón, A. Khanna and C. K. Jaggi, Impact of trade credit and inflation on retailer’s ordering policies for non-instantaneous deteriorating items in a two-warehouse environment, *International Journal of Production Economics*, 176 (2016), 154–169.

[19] Q. Wang, J. Wu, N. Zhao and Q. Zhu, Inventory control and supply chain management: A green growth perspective, *Resources, Conservation and Recycling*, 145 (2019), 78–85.

[20] J. Wu, F. B. Al-Khateeb, J.-T. Teng and L. E. Cárdenas-Barrón, Inventory models for deteriorating items with maximum lifetime under downstream partial trade credits to credit-risk customers by discounted cash-flow analysis, *International Journal of Production Economics*, 171 (2016), 105–115.

[21] J. Wu, L.-Y. Ouyang, L. E. Cárdenas-Barrón and S. K. Goyal, Optimal credit period and lot size for deteriorating items with expiration dates under two-level trade credit financing, *European Journal of Operational Research*, 237 (2014), 898–908.

[22] J. Zhang, G. Liu, Q. Zhang and Z. Bai, Coordinating a supply chain for deteriorating items with a revenue sharing and cooperative investment contract, *Omega*, 56 (2015), 37–49.

Received May 2020; revised October 2020.

E-mail address: Mahdi Karimi@ind.iust.ac.ir
E-mail address: sjsadjadi@iust.ac.ir