International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

ON PROGRAMMING DEVELOPMENT ON INVENTORY PROBLEM USING VISUAL BASIC

BY

Adeoye Akeem.O and Alawaye A.I

[1,2]Department of Mathematics and Statistics Federal Polytechnic Offa

---

Abstract - *The aim of this research is to develop a programme for inventory model on three method, when shortage is allowed, when shotage is not allowed and replenishment modeld. Visual basic was used to write the programme on the tree method. Result of the programm shows that the programm is efficient to solve any problem in inventory that involve shortage is allowed, when shortage is not allowed and replenishment model.*

INTRODUCTION

Inventory means a stock, and stock is a significant portion of business assets, which requires substantial investments. Inventory or stock includes raw materials; semi finished (work in progress) goods, and finished goods.

(i) Raw materials are purchased from external supplies.

(ii) Semi finished goods may be bought from a supplier or manufactured within the company/ organization itself, while

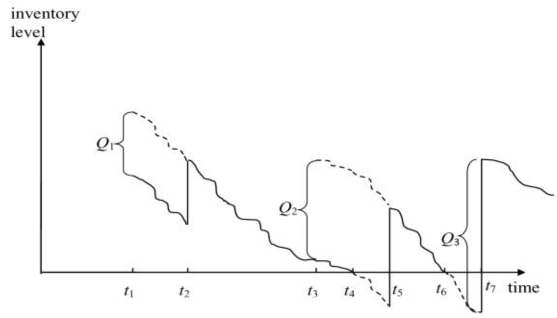
(iii) Finished goods (expected in the retail) are manufactured internally.

It is important that stocks are managed efficiently to prevent the investment becoming unnecessarily large. At the same time, stocks must be available to meet the demand for the consumption. For many organizations, management of inventories is of crucial importance. Inventory models will concern just a *single item* for which the demand per period (year) is D units. The number of items in stock is depleted over time by the demand, but also increased from time to time by instantaneous additions Q called *orders*, resulting in sudden jumps in the inventory level. The time between two consecutive replenishments is the *inventory cycle length* t_c.The *lead time* t_L is the time between the placement and arrival of an order.Three types of inventory/stock levels:

I^O , *inventory on hand* is stock physically on the shelf, immediately available to satisfy demand. $I^O \geq 0$.

I^N , *net inventory* on hand, is inventory on hand minus *backorders* (unsatisfied demand). $I^N = I^O -$ (backorders) may be negative if backorders exceed inventory on hand.

I^P, *inventory position* is $I^P = I^N +$ (outstanding orders).



Order size Q and the lead time t_L will usually be the same for all orders. The cost for carrying one unit of inventory for one period is called the unit *carrying* or *holding cost* c_h. For some inventory models, backorders are not allowed, otherwise the unit *shortage cost* c_s is the cost charged to be out of stock by one unit for one period of time. *Ordering cost* c_o is the cost of placing an order and having it delivered, considered independent of the size of the order.

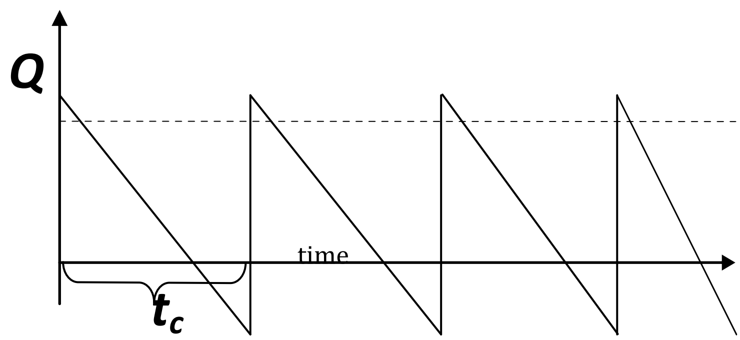
The Economic Order Quantity (EOQ) Model

Assumptions of the basic EOQ model are:

- Inventory is one single unperishable good,
- the demand rate is constant over time,
- the same amount is ordered each time,
- there are no quantity discounts,
- stockouts are not allowed,
- the planning horizon is infinite.

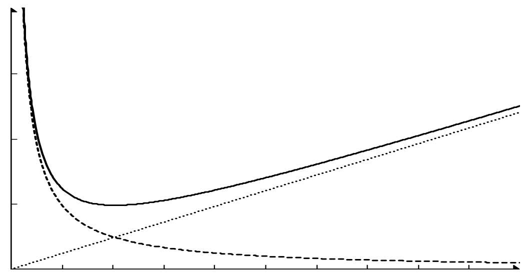
The famous sawtooth pattern is shown in the figure below.

inventory level



Ordering costs: if we place N orders, each of size Q, the total amount ordered is $NQ = D$, so that $N = D/Q$ is the number of orders per period and total annual ordering costs are $c_oN = c_oD/Q$. For *holding cost*, we compute the total area under the saw tooth curve, which can be seen to be ½Q, so that the total annual holding cost is c_h(½Q). The total inventory costs TC per period are

$$TC = c_o D/Q + ½ c_h Q$$



The value Q^* that minimizes total inventory cost TC, is the optimum order quantity

$$Q^* = \sqrt{\frac{2Dc_o}{c_h}}$$

$$TC(Q^*) = \sqrt{½Dc_oc_h} + \sqrt{½Dc_oc_h} = \sqrt{2Dc_oc_h}.$$

Note that holding and ordering costs are equal at optimum regardless of the value of the parameters D, c_o, and c_h.

METHODOLOGY

VISUAL BASIC SOURCE CODE

```
Private Sub Command1_Click()
If Combo1.Text = "Shortage Allowed" Then
frmShortage.Show
ElseIf Combo1.Text = "Replenishment" Then
frmReplenishment.Show
Else
frmNot.Show
End If
End Sub

Dim A, B, C, D, k, h, Q, TC, T, N, r, Re As Double
Private Sub CmdSN_Click()
k = Val(txtK.Text)
h = Val(txtH.Text)
r = Val(txtR.Text)
Q = Sqr((2 * k * r) / h)
lblEco.Caption = Round(Q, 4)
TC = Sqr(2 * h * k * r)
lblAso.Caption = Round(TC, 4)
T = Q / r
lbllen.Caption = Round(T, 4)
N = r / Q
lblexp.Caption = Round(N, 4)
End Sub

Private Sub cmdR_Click()
k = Val(txtK.Text)
h = Val(txtH.Text)
r = Val(txtR.Text)
Re = Val(txtRe.Text)
Q = Sqr((2 * k * r) / (h * (1 - (r / Re))))
lblEco.Caption = Round(Q, 4)
TC = Sqr((2 * k * h * r) * (1 - (r / Re)))
lblAso.Caption = Round(TC, 4)
T = Q / r
lbllen.Caption = Round(T, 4)
N = r / Q
```

```
lblexp.Caption = Round(N, 4)

End Sub


Dim k, h, Q, W, Q1, T, N, r, Re As Double

Private Sub CmdSA_Click()

k = Val(txtK.Text)

h = Val(txtH.Text)

r = Val(txtR.Text)

p = Val(txtP.Text)

Q = Sqr((2 * k * r) / h)

lblEco.Caption = Round(Q, 4)

Q1 = (Sqr((2 * k * r) * (p + h) / (p * h)))

lblAso.Caption = Round(Q1, 4)

W = (Sqr(2 * k * r * h) / (p * (p + h)))

lbllen.Caption = Round(W, 4)

End Sub
```

CONCLUSION

The programm developed can be use to solve any problem involve when shortage is allowed, when shortage is not allowed and replenishment model.

Reference

Abel Mizahr and Michael Sillivant (1979): Mathematics for Business and Social Sciences Published by John Willey & Sons Inc.

Don't Phillips A.R Arvndra and James J. Solberge (1976): Operation Research Principle and Practice Published by John Willey & Sons.

Pal Inc (1991): Operation Research Techniques Published by Phenum Press New York.

J.K. Lenstra, A.H.G. Rinnooy Kan, A. Schrijver (editions) History of Mathematical Programming: a collection of personal reminiscences, North-Holland, Amsterdam, 1991.

Mark de Berg, Mare van Kreveld, Mark Overmars, and Otfried Schwarzkopf (2000). Computational Geometry (2nd revised ed.). Springer-Verlag.

V. Chandru and M.R. Rao (1999) Linear Programming, Algorithms and Theory of Computation CRC Press.

(http://www.cse.iitb.ac.in/dbms/Data/Papers-Other/Algorithms/1p.ps.gz).

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein 2012 Introduction to Algorithms, fourth Edition. MIT Press and McGraw-Hill.

Michael R. Garey and David S. Johnson (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman.

Bernd Gartner, Jiri Matousek (2006). Understanding and Using Linear Programming.

Jalaluddin Abdullah, Optimization by the Fixed-Pont Method, Version 2.014.[3] (http://www.optimizationonline.org/DB_HTML/2007/09/1775.html).

Alexander Schrijver (1998) Theory of Linear and Integer Programming. John Wiley & Sons.

Michael. J. Todd (February 2002). "The many facets of linear programming" Mathematical Programming 91 (3).(3).

Vazirani, Vijay V. (2001). Approximation Algorithms. Springer-Verlag.