A STUDY OF STOCHASTIC MODEL OF INVENTRY WITH A COMPUTER SUPPORT

ABUBAKAR, Usman Yusuf & HAKIMI, Danladi
Department of Mathematics/Statistics,
Federal University of Technology Minna, Nigeria.
E-mail: hakimi_shengu@yahoo.com
Phone No: +234-803-451-3313

Abstract

The probabilistic inventory model studied in this paper introduces uncertainty in the aspect of lead time and reorder point. It was further considered for back orders and lost sales cases. The lead time and reorder point were described by a normal distribution function for the sake of simplicity. The model led to iterative procedure yet cumbersome to manipulate. Consequently, a simple computer program in Java scripts was designed to achieve the computation. The software was tested on a known problem and a known solution. It returned the exact solution in about a second; and with the values of the reorder point and the order quantity determined by the program, then the estimation of the expected annual cost (EAC) is almost trivial. The difficulty that is associated with the use of the model is thereby reduced. The model can be used to study inventory related problems such as the quantity of meal to be served to the passengers of an aircraft.

Keywords; Inventory, Order quantity, Lead time, Back-orders, Lost sales, Expected annual cost.

Introdunction

One class of inventory problems requires that the order quantity decision be made only once for the entire demand process. That is, the opportunity to replenish the stock as it becomes depleted does not occur. The major distinguishing characteristic is that the order quantity decision is a "one shot" affair and that, despite uncertainty in the demand, a proper trade-off must be achieved between the consequences of too much and too little. The model can be used to solve many hypothetical inventory problems such as: How many fresh bread to bake for a given day? Some of the recent researches reported in literature include, Abou et al (2002) they studied probabilistic multi-item inventory model with varying order cost. Also, the optimal lot size for the finite production model with random defective rate and back logging, and a numerical method for determination of reworking or scraping the defective items in a finite rate model were contained in Chiu (2003) and Chiu et al (2006) respectively. Production lot sizing with the reworking of imperfect quantity items produced was reported in Hayek et al (2001). In this work, a study is made of a single period, probabilistic, lot size model. A computer software was provided to facilitate the calculations of the identified cumbersome iterative procedure for easy and wider application of the model.

The Model

The model discussed below is cited from Korve(1994). It is assumed that the status of the inventory is known at all times and the system is one of continuous review. There will be a lead time for replenishment, denoted by *L*. we want to consider the effect of making *L* a random variable. The policy assumed for this model is that the replenishment order will be initiated the moment that the inventory level falls to a certain value called the reorder point, denoted *r*. The value of *r* is one of the quantities to be determined.

The other decision variable is Q, the order quantity. Because orders always occur when the inventory level is r and everything else is the same. Regardless of what L, r and Q may be, there will be a positive probability that the inventory will be depleted before the lead time expires. Consequently, what happens when this occurs must be specified.

There are two obvious choices: demands that occur when the stock is out and are accumulated as back orders and fulfilled when the order arrives, or such demands are simply never fulfilled. The former case is called the Back-Orders Case; the latter is referred to as the Lost-Sales case. Both are treated here. In either case, p will be used as the penalty cost per item. That is, if the inventory is exhausted, each unit demanded until the stock is replenishment will cost p Naira.

The first step in the formulation of the expected annual-cost function is to identify the separate components of cost. Accordingly, let

$$EAC(Q,r) = OC + SC + HC$$

Where EAC(Q,r) stands for Expected Annual cost, as a function of the decision variables Q and r, OC stands for Ordering Cost, SC for stock out cost, and HC for Holding Cost. The terms on the right can now be developed separately.

The ordering cost is quite easy to obtain. It is a, the cost per order, times the expected number of cycles per year. Since d is the expected demand per year and Q is the amount Sold per cycle, the expected number of cycles per year is d/Q, provided that all demand is met. So, at least in the back-orders case,

$$OC \quad a\frac{d}{Q}$$

In the lost-sales case, this expression is not exactly correct. If we assume, however, that p is sufficiently large to ensure that the policy ultimately used will result in few lost sales, the same expression will suffice as a close approximation to the expected ordering cost even in the lost-sales case.

The back-orders case.
The cycle begins when an order arrives. About the inventory level, it would be r when this arriving order was initiated. Since that time, for the period L, demand has continued in its random fashion. The expected demand during L is μ, so the expected inventory level just before the order arrives is $r - \mu$ (which may conceivably be negative), and the expected inventory level at the start of our cycle is $Q+r-\mu$. The demand process then depletes the inventory, another replenishment order is placed, and the cycle ends with an expected inventory level of $r - \mu$. The average inventory over this cycle is midway between $Q+ r - \mu$ and $r -\mu$, or

$$\frac{Q \quad r - \quad r -}{2} \quad \frac{Q}{2} \quad r -$$

By this argument, the expected annual and holding cost expression is

$$HC \quad \square \quad - \quad r -$$

But this argument was for the back-orders case, however, the average inventory in the lost-sales case will be slightly higher than in the back orders case. The expected inventory level just before the order arrives is not quite $r - \mu$, because the inventory is not allowed to go negative. To see what it is instead, let x be the demand during the lead time L. Then the inventory level is

$$r - x; \quad \text{if } x \le r$$
$$0; \quad \text{if } x \ge r$$

Taking the expected value of this function of x gives for the expected inventory level

$$\int r - x\, f\, x\, dx \quad \int_{\infty} 0 f\, x\, dx$$

$$\int_{\infty} r - x\, f\, x\, dx - \int_{\infty} r - x\, f\, x\, dx$$

$$r \int^{\infty} f\, x\, dx - x \int^{\infty} f\, x\, dx \quad \int^{\infty} r - x\, f\, x\, dx$$

$$r - \quad B\, r$$

In words the expected inventory level just before replenishment is greater by an amount equal to the expected number of lost sales. The average inventory can be found as in the back-orders case. It works out to give for the expected annual holding cost in the lost-sales case.

$$HC \quad \frac{Q}{2} \quad r - \quad B\, r$$

The total expected annual cost is therefore

$$EAC\, Q,r \quad \frac{ad}{Q} \quad \frac{pd}{Q} B\, r \quad \frac{Q}{2} \quad r -$$

in the back-orders case, and the same expression plus *hB(r)* in the lost-sales case. This is the function of *Q* and *r* which is to be minimized; that is, the order quantity and reorder point that yield the smallest expected annual cost to be found. The minimum will occur for values of *Q* and *r* that are nonnegative and finite, and will occur at that point where the partial derivatives with respect to *Q* and *r* both vanish.

So, taking the partial derivative with respect to *Q*,

$$\frac{\delta EAC\, Q,r}{\delta Q} \quad -\frac{ad}{Q} - \frac{pdB\, r}{Q} \quad \frac{}{2}$$

Setting it equal to zero and solving for *Q*, we find

$$Q \quad \sqrt{\frac{2d\, a \quad pB\, r}{}}$$

Note the similarity of this expression to the classical, deterministic economic order quantity formula. It appears that the optimal-order quantity when demand is probabilistic is the same as it would be in the deterministic case if *a* were increased by the amount *pB(r)*. Since *p* is the cost per back order or lost sale and *B(r)* is the expected number of back orders or lost sales per cycle, *pB(r)* is the expected cost attributable to stock outs per cycle. If you think about it, the addition of this quantity to *a* makes some sense. Of course, *r* and *B(r)* are still unknown.

Returning, then, to the expected annual cost function and taking the partial derivate with respect to *r*, we get

$$\frac{\delta EAC\, Q,r}{\delta r} \quad \frac{pd}{Q} \frac{dB\, r}{dr}$$

In the back-orders case, or

$$\frac{\delta EAC\, Q,r}{\delta r} \quad \frac{pd}{Q} \quad \frac{dB\, r}{dr}$$

In the lost-sales case, recalling that

$$B\ r\qquad \int^{\infty}\ r - x\ f\ x\ dx$$

and making use of Leibniz's rule to differentiate the integral.

$$\frac{dB\ r}{dr}\quad - \int^{\infty}\ f\ x\ dx$$

Substituting this back into the partial derivative, setting it equal to zero, and solving, we get

$$\int^{\infty}\ f\ x\ dx\qquad \frac{Q}{pd}$$

in the back-orders case, or

$$\int^{\infty}\ f\ x\ dx\qquad \frac{Q}{pd\quad Q}$$

in the lost-sales case.

The best expression for *r* that we can obtain is a critical ratio solution (the last two equations). The left-hand side is the complementary cumulative distribution of lead-time demand evaluated at *r*, or the probability that the demand during *L* exceeds *r*. In words, the reorder point *r* must be set high enough so that the probability of running out is just equal to the critical ratio. For example, if *p* is very large, the critical ratio will tend to be small, which will in turn force the reorder point to be large enough to provide a generous margin of safety against stock outs.

Although we now have expressions for both *Q* and *r*, the problem is not solved because the evaluation of each of the expressions requires knowledge of the value of the other. To get *Q* we need *r* and to *r* we need *Q*. Fortunately, there is an iterative approach that works out quite nicely in practice.

The Iterative Approach

1. As an initial, temporary value, let *B(r)* = 0, and solve for

$$Q\qquad \frac{2d\ a\quad pB\ r}{}$$

This would amount to solving for the deterministic optimal-order quantity

2. Using Q_1, find *r,* from the critical ratio rule:

$$\int^{\infty}\ f\ x\ dx\qquad \frac{Q}{pd}\quad ....\qquad b\quad - order\qquad e$$

$$\frac{Q}{pd\quad Q}\quad ....\qquad o -\qquad e\qquad e$$

3. Using r_1, evaluate

$$B\ r\qquad \int^{\infty}\ x - r\ f\ x\ dx$$

4. Using *B(r₁)*, solve for Q_2 as in step 1.
5. Using Q_2, find r_2 as in step 2.
6. Continue in this manner until no change occurs in the values of Q_i and r_i. The final values obtained are the optimal-order quality and reorder point.

In practice, convergence is usually rapid. Often Q_2 and r_2 will be so close to optimal that further iterations are not worth the trouble.

The solution procedure required for this model is cumbersome. Although it would pose no difficulty for implementation as a computer algorithm, it is necessary to evaluate the integral involved in *B(r)*

Recall that

$$B\ r \quad \int^{\infty} \ x - r\ f\ x\ dx$$

where *f(x)* is the lead-time demand density function.   When *f(x)* is complicated, as it may reasonably be expected to be most of the time, the integration may not be trivial. Perhaps the most important special case occurs when the lead-time demand distribution is normal, in which case.

$$B\ r \quad \int^{\infty} \ x - r\ \frac{1}{\delta\sqrt{2}}exp \ -\frac{1}{2}\ \frac{x-}{\delta} \quad dx$$

Fortunately this rather imposing expression can be translated to an expression involving only the standard normal density and cumulative distribution functions, which are readily available in tables. After some preliminary manipulation,

$$B\ r \quad \int^{\infty} \ x - \quad - r\ \frac{1}{\delta\sqrt{2}}exp \ -\frac{1}{2}\ \frac{x-}{\delta} \quad dx$$

$$\int^{\infty} \frac{x-}{\delta} \quad \frac{x-}{\delta}\ \frac{1}{\sqrt{2}}exp \ -\frac{1}{2}\ \frac{x-}{\delta} \quad dx$$

a change of variable from x to y $x - \ /\delta$ produce

$$B\ r \quad \int^{\infty}_{/} \frac{1}{\sqrt{2}}e^{-}\ \delta d \quad \int^{\infty}_{/} \frac{-r}{\delta}\ \frac{1}{\sqrt{2}}e^{-}\ \delta d$$

$$\delta \int^{\infty}_{/} \frac{1}{\sqrt{2}}e^{-}\ d \quad - r \int^{\infty}_{/} \frac{1}{\sqrt{2}}e^{-}\ d$$

Now the first integral is easy to evaluate

$$\int^{\infty}_{/} \frac{1}{\sqrt{2}}e^{-}\ d \quad \frac{1}{\sqrt{2}} \int^{\infty}_{/} \ e^{-}\ d$$

$$\overline{\sqrt{}}\ e^{-}\ |^{\infty}\ /$$

$$\overline{\sqrt{}}\ exp \ -- \ \overline{\quad} \quad f \ \overline{\quad}$$

where *f(x)* is the standard normal density function. The second integral is even easier; it is already tabulated as the complementary cumulative for the standard normal. In summary,

$$B\ r \quad \delta f\ \frac{r-}{\delta} \quad - r\ \frac{r-}{\delta}$$

Where G*(x)* is the complementary cumulative distribution function for the standard normal, (Just to be sure that there is no confusion; the lower case *f* refers to the height of the density function, and the upper case *G* referred to the area under the right-hand tail of the density function). Of course, this expression is valid only when the lead –time-demand distribution is normal. The assumption that lead-time demand is in fact normal is frequently justified by the Central Limit Theorem. The total demand can be written as a sum of the demands of individual customers, so if there are enough of them, the total should approach a normally distributed random variable.   There are some weaknesses in this argument, but it provides some reassurance that the normality assumption is not completely arbitrary, in many real-world problems.

The Program Development
A computer program shall now be developed to achieve the calculations identified in the previous section.

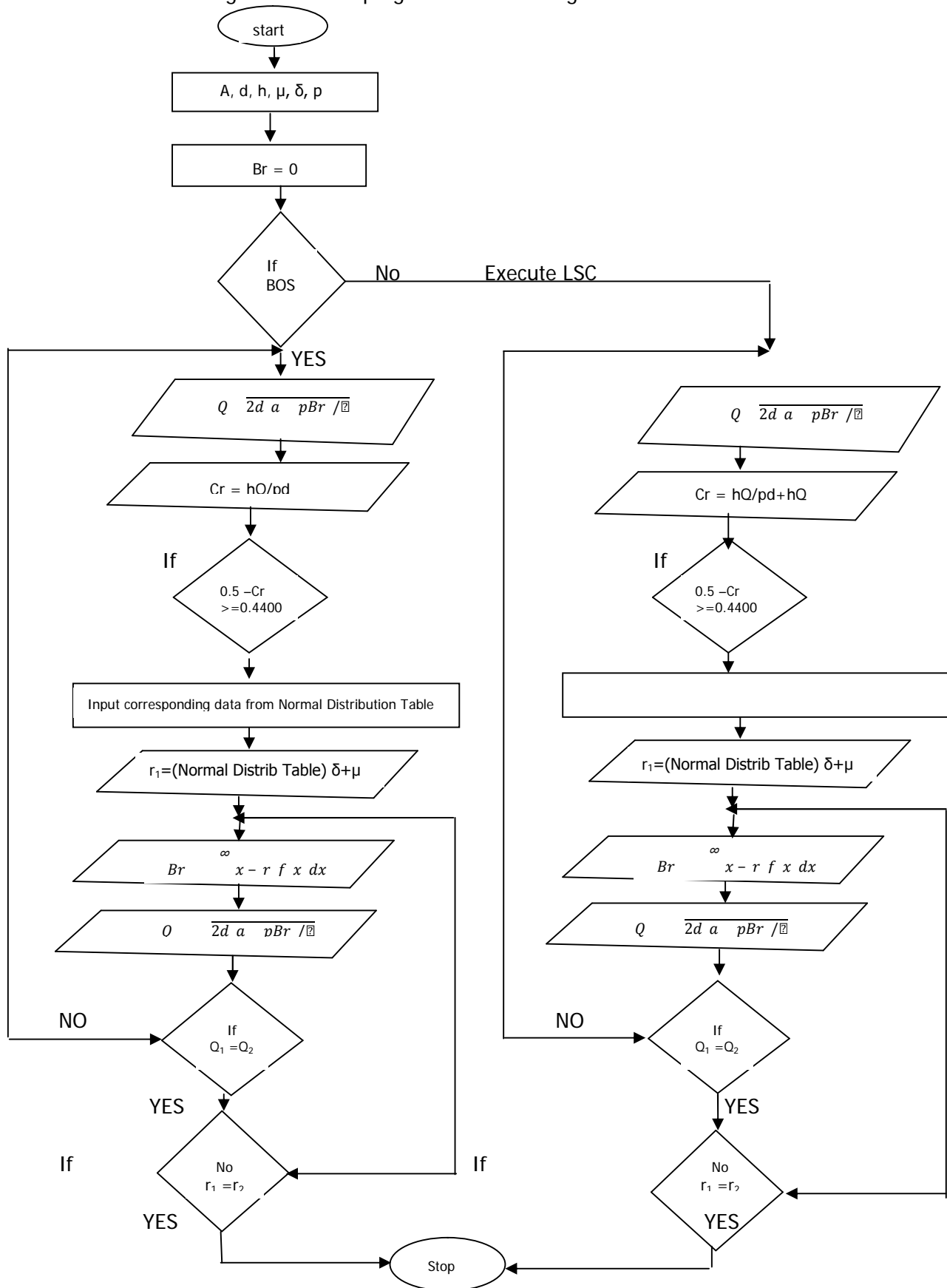Flowchart. The algorithm of the program is shown in figure 1.

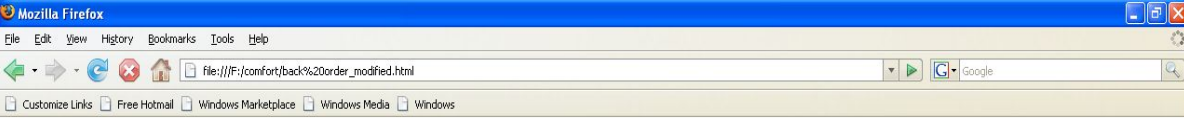Figure 1: The Algorithm of the program in a flowchart

Program Code
Program code in Java Script is shown in Appendix 'A'.

Results
For the System Implementation, the following problem was tested on the software and the output is presented in table1 and table2 respectively. Suppose that a large department store sells blank recording tape under its own brand name. The tape itself is identical to that of a well-known recording tape manufacturer, and is in fact supplied by that manufacturer. Because of the special labeling and packaging, however, there is a lead time of 5.2 weeks, or 1/10 year. Assume that the demand during a period of this length is normally distributed with a mean of 1000 tapes and a standard deviation of 250. By implication, the mean annual demand must be 10,000 tapes. The cost of paper work and handling associated with placing an order is ₦100, and the holding cost is 15% tape per year. Although we shall want to see what happens in the lost-sales case as well, assume for now that back orders are taken. The penalty is assumed to be ₦1.00 per tape back ordered, Korve(1994).

Table1. The result for back order case

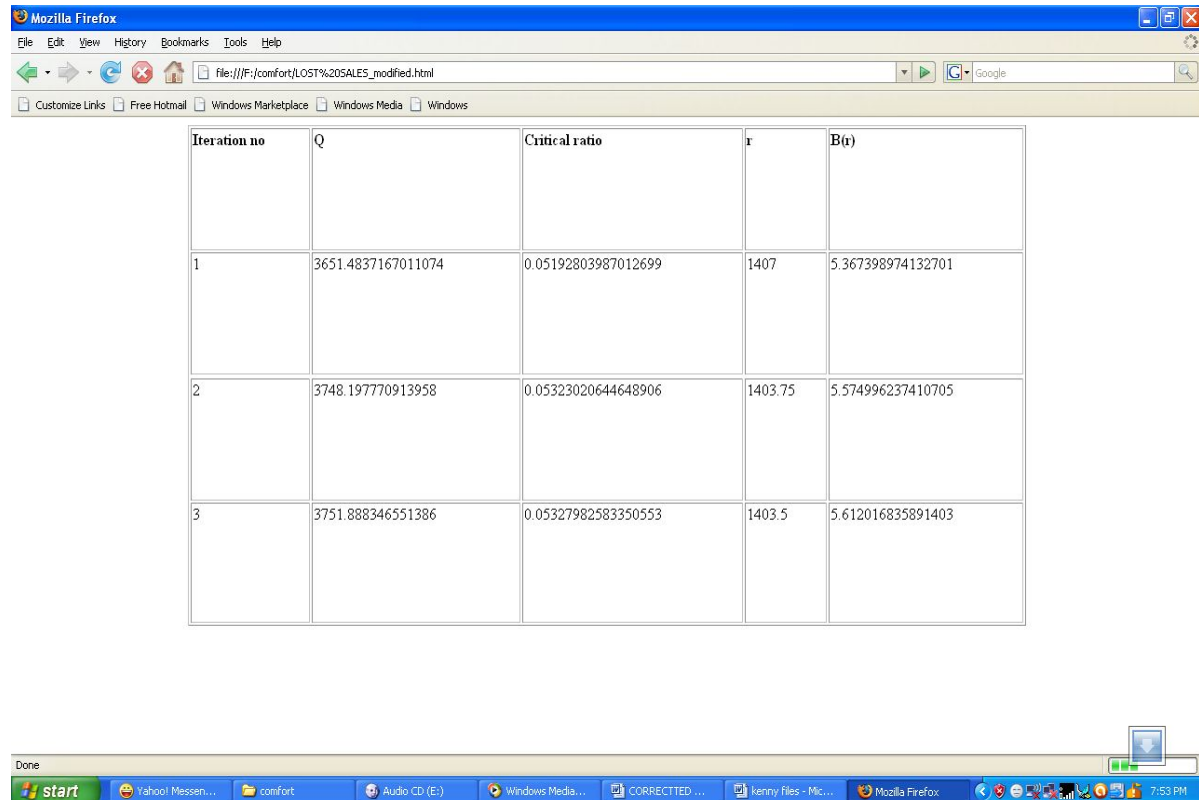| Iteration no	Q	Critical ratio	r	B(r)
1	3651.4837167011074	0.054772255750516606	1400	5.818367487488139
2	3756.2102796566674	0.05634315419485001	1396.75	5.953498241459499
3	3758.6078671312957	0.05637911800696943	1396.75	5.93922959900112

Table 2: The result for lost sale case

Iteration no	Q	Critical ratio	r	B(r)
1	3651.4837167011074	0.05192803987012699	1407	5.367398974132701
2	3748.197770913958	0.05323020644648906	1403.75	5.574996237410705
3	3751.888346551386	0.05327982583350553	1403.5	5.612016835891403

Conclusion

The probabilistic model considered in this work introduces the uncertainty in the aspect of lead time and reorder point. It was described by a normal distribution for the sake of simplicity. The model led to iterative procedure yet cumbersome to manipulate. Consequently, a simple computer program in Java Scripts was designed to achieve the computation. The software was tested on a known problem and a known solution. It returned the exact solution in about a second presented in table1 and table2 for the back orders case and lost sales case respectively. The program is interactive and the required values are entered from the standard normal table while the program runs. With the values of B(r) (the reorder point) and Q (order quantity) contained in table1 and table2, the estimation of the expected annual cost (EAC) becomes very simple. The difficulty that is associated with the use of the model is thus reduced. The model could be used to study inventory related problems such as the quantity of meal to be served to the passengers of an air craft.

References

Abou-El-Ata, F., H., A. & El-Wakeel, M. F. (2002). Probabilistic multi-item inventory model with varying order cost under two restrictions: A geometric programming approach. *Intl. J. Product. Econ.,* 83, 223-231.

Chiu, P. Y. (2003). Determining the optimal lot size for the finite production model with random defective rate, the rework process and backlogging. *Engineering Optimization,* 35 (4), 427-437.

Chiu, Y. P., Chiu, S. W. & Chao, H. (2006). Numerical method for determination of reworking or scraping the defective items in a finite production rate model. *Communications in Numerical Methods in Engineering,* 22(5), 377-386.

Goyal, S. K. (1980). A note on the paper: An inventory model with finite horizon and price changes. *J. Opnl. Res. Soc.,* 30 (9).

Hayek, P. A. & Salameh, M. K. (2001). Production lot sizing with the reworking of imperfect quality items produced. *Production Planning and Control,* 12 (6), 584-590.

Korve, K. N. (1994). *Operations research MAT537 (Inventory Theory).* Unpublished lecture notes. Department of Mathematics, University of Jos Nigeria.

APPENDIX 'A'

SOURCE CODE:

```
<html>
<CENTER><h1>BACK ORDER CASE </h1></CENTER><br><br>
<script language=javascript>
var p,d,a,h,u,sd,g;
function gofr(){
p=document.joe.easy.value;
d=document.joe.tooeasy.value;
a=document.joe.asol.value;
h=document.joe.hsol.value;
u=document.joe.usol.value;
sd=document.joe.sdsol.value;
p=parseInt(p);
d=parseInt(d);
a=parseInt(a);
h=parseFloat(h);
u=parseInt(u);
sd=parseInt(sd);
gw=Math.sqrt(2*d*a/h);
document.joe.qgteasy.value=gw;
}   //q1 result
function sec(){
crans=(h*gw)/(p*d);
document.joe.cr.value=crans;}  //cr1 result
function tg(){
fftv=0.5-crans;
document.joe.toomuch.value=fftv
} //four figure table value
function mata(){
nb=document.joe.ktst.value;
g=(nb*sd)+u
document.joe.mp.value=g}   //r1 result
function fb(){
wwe=(-0.5)*(nb*nb);
hee=Math.pow(Math.E,wwe);
genius=(u-g)*crans;
ebr=sd*(0.3989*hee)+genius;
document.joe.brte.value=ebr}  //br1 result
function qtwoo(){
qii=Math.sqrt((2*d*(a+ebr))/h)
document.joe.qtwo.value=qii
}                              //q2 result
function crtwoo(){
crii=(h*qii)/(p*d)
document.joe.crtwo.value=crii}//cr2 result
function fftva()
```

```
{fii=0.5-crii
document.joe.fftvt.value=fii} //four figure table value two
function rtwo(){
rsfu=document.joe.fftrii.value
rtt=(rsfu*sd)+u
document.joe.rii.value=rtt}//R2 result
function brtwo(){
bre=(-0.5)*(rsfu*rsfu)
bro=(u-rtt)*crii
brf=Math.pow(Math.E,bre)
brse=sd*(0.3989*brf)+bro
document.joe.brii.value=brse}//Br2 result
function qthreeo(){
qiii=Math.sqrt((2*d*(a+brse))/h)
document.joe.qthree.value=qiii
}                                    //Q3 result
function crthree(){
criii=(h*qiii)/(p*d)
document.joe.crtiii.value=criii}           //Cr3 result
function ftr(){
lass=0.5-criii
document.joe.gtow.value=lass}
function abf(){
mla=document.joe.idtims.value
itsover=(mla*sd)+u
document.joe.ntr.value=itsover}       //R3 result
function brov(){
bron=(-0.5)*(mla*mla)
brp=(u-itsover)*criii
bpif=Math.pow(Math.E,bron)
finishedit=sd*(0.3989*bpif)+brp
document.joe.rtcb.value=finishedit}      //Br3 result
function winner(){
document.write("<center>","<table      width=900      height=500      border=1>","<tr>","<td
valign=top>","<b>","Iteration                                no","</b>","</td>","<td
valign=top>","<b>","Q","</b>","</td>","<td                           valign=top>","<b>","Critical
ratio","</b>","</td>","<td                      valign=top>","<b>","r","</b>","</td>","<td
valign=top>","<b>","B(r)","</b>","</td>","</tr>","<tr>","<td   valign=top>","1","</td>","<td
valign=top>",gw,"</td>","<td   valign=top>",crans,"</td>","<td   valign=top>",g,"</td>","<td
valign=top>",ebr,"</td>","</tr>","</tr>","<tr>","<td    valign=top>","2","</td>","<td
valign=top>",qii,"</td>","<td    valign=top>",crii,"</td>","<td    valign=top>",rtt,"</td>","<td
valign=top>",brse,"</td>","</tr>","<tr>","<td                    valign=top>","3","</td>","<td
valign=top>",qiii,"</td                valign=top>","<td              valign=top>",criii,"</td>","<td
valign=top>",itsover,"</td>","<td
valign=top>",finishedit,"</td>","</tr>","</table>","</center>")
}
function remedy(){
```

```
champ=window.open("","pl","scrollbars=yes,width=700,height=700,resizable=yes,menubar=no,
toolbar=no")
champ.document.write("<center>","<table   width=700   height=500   border=1>","<tr>","<td
valign=top>","<b>","Iteration                                      no","</b>","</td>","<td
valign=top>","<b>","Q","</b>","</td>","<td          valign=top          >","<b>","Critical
ratio","</b>","</td>","<td                          valign=top>","<b>","r","</b>","</td>","<td
valign=top>","<b>","B(r)","</b>","</td>","</tr>","<tr>","<td   valign=top>","1","</td>","<td
valign=top>",gw,"</td>","<td  valign=top>",crans,"</td>","<td  valign=top>",g,"</td>","<td
valign=top>",ebr,"</td>","</tr>","<tr>","</tr>","<tr>","<td    valign=top>","2","</td>","<td
valign=top>",qii,"</td>","<td    valign=top>",crii,"</td>","<td    valign=top>",rtt,"</td>","<td
valign=top>",brse,"</td>","</tr>","<tr>","<td                valign=top>","3","</td>","<td
valign=top>",qiii,"</td         valign=top>","<td          valign=top>",criii,"</td>","<td
valign=top>",itsover,"</td>","<td
valign=top>",finishedit,"</td>","</tr>","</table>","</center>","<br>")}
</script>
<CENTER>
<form name=joe>
<i>input value for <b>p</b> <input type=text  name=easy>  input value for <b>d</b>
<input type=text name=tooeasy>
input value for <b>a</b>
<input type=text name=asol><br><br>
input value for <b>h</b>
<input type=text name=hsol>
input value for <b>u</b>
<input type=text name=usol>
input value for <b>&#8706</b>
<input type=text name=sdsol></CENTER><br><br><br><br>
<center><b>for the first iteration Br=0</b>
<br><br>
<b>Q1=&#8730(2d(a+p(br)/h))</b><br>
<b>Q1</b> result appears here
<input     type=text     name=qgteasy><input    type=button    value="calculate    Q1"
onClick=gofr()><br><br>
<b>Cr1=hQ1/pd</b><br>
<b>Cr1</b> result appears here
<input      type=text      name=cr><input      type=button      value="calculate      Cr1"
onClick=sec()><br><br>
<b>0.5-Cr1</b><br>
check this result in four figure table
<input type=text name=toomuch><input type=button value=calculate onClick=tg()><br><br>
input result from four figure table here
<input type=text name=ktst><br><br>
<b>(R1-u/&#8706)=four figure table result</b><br>
<b>R1</b> result appears here
<input      type=text      name=mp><input      type=button      value="calculate      r1"
onClick=mata()><br><br>
<b>Br1=&#8747<sub>r1</sub><sup>&#8734</sup>(x-r1) f(x)dx</b><br>
<b>Br1</b> result appears here
```

```
<input type=text name=brte><input type=button value="calculate Br1" onClick=fb()>
<br><br><br><br>
<b>for the second iteration</b><br><br>
<b>Q2=&#8730(2d(a+p(br1)/h))</b><br>
<b>Q2</b> result appears here
<input       type       text       name=qtwo><input       type=button       value="calculate       Q2"
onClick=qtwoo()><br><br>
<b>Cr2=hQ2/pd</b><br>
<b>Cr2</b> result appears here
<input       type       text       name=crtwo><input       type=button       value="calculate       Cr2"
onClick=crtwoo()><br><br>
<b>0.5-Cr2</b><br>
check this result in four figure table
<input type text name=fftvt><input type=button value=calculate onClick=fftva()><br><br>
input result from four figure table here
<input type text name=fftrii><br><br>
<b>(R2-u/&#8706)=four figure table result</b><br>
<b>R2</b> result appears here
<input type text name=rii><input type=button value="calculate r2"onClick=rtwo()><br><br>
<b>Br2=&#8747<sub>r2</sub><sup>&#8734</sup>(x-r2) f(x)dx</b><br>
<b>Br2</b> result appears here
<input       type=text       name=brii><input       type=button       value="calculate
Br2"onClick=brtwo()><br><br><br><br>
<b>for the third iteration</b><br><br>
<b>Q3=&#8730(2d(a+p(br2)/h))</b><br>
<b>Q3</b>  result  appears  here<input  type=text  name=qthree><input  type=button
value="calculate Q3" onClick=qthreeo()><br><br>
<b>Cr3=hQ3/pd</b><br>
<b>Cr3</b>  result  appears  here<input  type=text  name=crtiii><input  type=button
value="calculate Cr3" onClick=crthree()><br><br>
<b>0.5-Cr3</b><br>
check this result in four figure table
<input type=text name=gtow><input type=button value=calculate onClick=ftr()><br><br>
input result from four figure table here<input type=text name=idtims><br><br>
<b>(R3-u/&#8706)=four figure table result</b><br>
<b>R3</b> result appears here
<input type=text name=ntr><input type=button value="calculate r3"onClick=abf()><br><br>
<b>Br3=&#8747<sub>r3</sub><sup>&#8734</sup>(x-r3) f(x)dx</b>
<br>
<b>Br3</b> result appears here
<input       type=text       name=rtcb><input       type=button       value="calculate       Br3"
onClick=brov()><br><br><br><br>
<input type=button value="generate single table" onClick=winner()>
<input type=button value="generate multiple tables" onClick=remedy()>
</form><br><br>
<font size=-1>
ss
```