Industrial Engineering Applications Using Microcomputers

Farid Guediri
University of Central Florida
INDUSTRIAL ENGINEERING APPLICATIONS
USING MICROCOMPUTERS

BY

FARID GUEDIRI
B.S., University of Central Florida, 1978

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program of the College of Engineering
at the University of Central Florida at Orlando, Florida

Winter Quarter
1981
ABSTRACT

Five computer programs have been developed for practical use by Industrial Engineers in a production environment. The programs cover the areas of forecasting, learning efficiency, production control, network analysis and optimum equipment replacement policies. Theory, complete program documentation and case example, for each program, are presented. Programs are written in an interactive Basic mode and have been tested on the Radio Shack TRS-80 and Apple II Plus systems.
A mes parents Said et Yamina, 
ma femme Kay Marie, 
et mon oncle Mahmoud Chibani
ACKNOWLEDGEMENTS

The author acknowledges the valuable motivation and help of Dr. Y. A. Hosni, Dr. H. I. Klee, Dr. D. G. Linton, and my Department Chairman, Dr. G. E. Whitehouse.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................ iii

INTRODUCTION ............................................. 1

Chapter
 I. SEASONAL FORECASTING USING WINTER'S SMOOTHING ........ 3
   Introduction ............................................ 3
   Theory ................................................... 3
   Program Documentation .................................. 8
   Program Listing ......................................... 10
   Case Example ........................................... 15

II. THE LEARNING CURVE .................................... 22
   Introduction ............................................ 22
   Theory ................................................... 22
   Program Documentation .................................. 25
   Program Listing ......................................... 27
   Case Example ........................................... 31

III. LINE OF BALANCE (LOB) .................................. 35
   Introduction ............................................ 35
   Theory ................................................... 35
   Program Documentation .................................. 39
   Program Listing ......................................... 41
   Case Example ........................................... 46

IV. MAXIMAL-FLOW PROBLEM ................................. 58
   Introduction ............................................ 58
   Theory ................................................... 58
   Program Documentation .................................. 67
   Program Listing ......................................... 69
   Case Example ........................................... 72

V. EQUIPMENT REPLACEMENT PROBLEM ....................... 74
   Introduction ............................................ 74
   Theory ................................................... 74
Program Documentation ....................... 77
Program Listing .............................. 81
Case Example ............................... 85

LIST OF REFERENCES .......................... 92
INTRODUCTION

With the advance of technology, the processing power reliability, and the speed of the microcomputer have grown, while its cost has dropped significantly. The low cost and operational versatility of the microcomputer make it cost-effective for a wide variety of data processing applications not economically feasible with larger, more expensive computers.

Practicing Industrial Engineers will find it advantageous to acquire an inexpensive microcomputer to solve medium to small size problems within their department. This report presents five problem-oriented computer packages designed for self-documented interactive use. The areas covered by the programs are: Seasonal forecasting using Winter's smoothing technique is presented in Chapter I. The program generates a forecast of future observations given a historical data series which exhibits both a trend and a seasonal behavior.

In Chapter II, the Learning Curve theory is used in a computer program to provide a management tool for measuring performance time decrease as a result of learning through repetition. Manpower requirements and future production cost estimates can also be generated by the program.

Chapter III presents a computerized version of the traditional graphical Line of Balance technique. The program can be used to
monitor the progress during the production cycle as well as identify the bottlenecks and delays in production schedule.

The Maximal Flow Problem is presented in Chapter IV. The function of the program is to determine the optimum flow between a source and a sink in a network, subject to capacity restrictions.

The program in Chapter V uses dynamic programming to solve the Equipment Replacement Problem. Its objective is to determine the optimum replacement policy for a piece of equipment over a number of years.

The usage of the programs does not require any programming background. They can be used without reference to a user's manual. Instructions are given on how to start the program, input data and select execution options. Computer memory space, for data storage, is allocated according to the size of the user's problem and is only limited by the capacity of the microcomputer in use.
CHAPTER I

SEASONAL FORECASTING USING WINTER'S SMOOTHING

Introduction

Forecasting problems are of considerable importance in an industrial environment. Forecasting methods such as moving averages, exponential smoothing, etc., are limited to time series that do not exhibit seasonal behavior. The method developed by Winters (Montgomery, and Johnson 1976) is a higher form of smoothing technique which takes into consideration trend and seasonal behavior.

In this chapter, theory, computer program and case example are presented.

Theory

Time series which consider trend and seasonal effects can be represented by the model:

\[ x_t = (b_1 + b_2 t)c_t \]

where \( x_t \) = the forecast in future period \( t \)

\( b_1 \) = the base signal, usually called the permanent component

\( b_2 \) = a linear trend component

\( c_t \) = a multiplicative seasonal factor

The length of a season is \( L \) periods, and the seasonal factors are defined so that they sum to the length of the season, that is,
The estimates of the trend and seasonal factors, at the end of any time period $T$, are denoted as $\hat{b}_2(T)$ and $\hat{c}_T(T)$, respectively. The permanent component will be established on a current-origin basis, its estimate at the end of period $T$ is denoted $\hat{a}_1(T)$. At the end of the current period $T$, after observing the realization for that period, $x_T$, the estimates $\hat{a}_1(T)$, $\hat{b}_2(T)$ and $\hat{c}_T(T)$ are updated according to Winter's method:

1. Updating of the permanent component estimate:

$$\hat{a}_1(T) = A \frac{x_T}{c_T(T-L)} + (1-A) \left[ \hat{a}_1(T-1) + \hat{b}_2(T-1) \right]$$  \hspace{1cm} (1-3)

where $A = a$ permanent component smoothing constant ($0<A<1$). Dividing $x_T$ by $c_T(T-L)$, which is the estimate of the seasonal factor for period $T$ computed one season ($L$ periods) ago, deseasonalizes the data so that only the trend component and the prior value of the permanent component enter into the updating process for $\hat{a}_1(T)$. This shifts the origin of time to the end of the current period.

2. Updating of the trend component estimate:

The estimate of the trend component is the smoothed difference between two successive estimates of the permanent component.

$$\hat{b}_2(T) = B \left[ \hat{a}_1(T) - \hat{a}_1(T-1) \right] + (1-B)\hat{b}_2(T-1)$$  \hspace{1cm} (1-4)

3. Updating of the seasonal factor estimate for period $T$:

The current observed seasonal variation (that is, $x_T/\hat{a}_1(T)$)
is smoothed with the estimate of the seasonal factor for period \( T \) computed \( L \) periods ago, which was the last opportunity to observe this portion of the seasonal pattern, to obtain a new estimate of the seasonal effect in period \( T \).

\[
\hat{c}_T(T) = \frac{x_T}{a_1(T)} + (1-G)\hat{c}_{T-L}(T-L)
\]

(1-5)

where \( G = \) Seasonal smoothing constant \((0<G<1)\).

The forecast for any future period \( T+h \) is,

\[
\hat{x}_{T+h}(T) = [\hat{a}_1(T) + \hat{b}_2(T)h]\hat{c}_{T+h}(T+h-L)
\]

(1-6)

Forecasts for time periods more distant than \( L \) can be made by reusing the appropriate seasonal factor.

Winter's method requires initial values of the parameters \( \hat{a}_1(0), \hat{b}_2(0) \) and \( \hat{c}_L(0) \) for \( t = 1, 2, \ldots, L \). Historical information can be used to provide some or all of the initial estimates. Several heuristic algorithms have been devised to utilize historical data for initial parameter estimation.

The initialization procedure used in this chapter is:

- Estimate the trend component by

\[
\hat{b}_2(0) = \frac{\bar{x}_m - \bar{x}_1}{(m-1)L}
\]

(1-7)

where \( \bar{x}_j, j = 1, 2, \ldots, m, \) denote the average of the observations during the \( j \)th season.

- Estimate the permanent component by

\[
\hat{a}_1(0) = \bar{x}_1 - \frac{L}{2} \hat{b}_2(0)
\]

(1-8)
- Estimate the seasonal factor by

\[ \hat{c}_t = \frac{x_i - [(L+1)/2-j]b_2(0)}{t = 1, 2, \ldots, mL} \]  (1-9)

where \( x_i \) is the average for a season corresponding to the \( t \) index, and \( j \) is the position of period \( t \) within the season (for example, if \( 1<t<L \), the \( i = 1 \), and if \( L + 1<t<2L \), then \( i = 2, \text{ etc.} \)).

Equation (1-9) will produce \( m \) estimates of the seasonal factor for each period. These should be averaged to produce a single estimate of the seasonal factor for each period within the season.

- Average the seasonal factors by

\[ \bar{c}_t = \frac{1}{m} \sum_{K=0}^{m-1} \hat{c}_{t+KL} \quad t = 1, 2, \ldots, L \]  (1-10)

where \( \bar{c}_t \) is an average seasonal factor for period \( t \).

Finally, the seasonal factors should be normalized so that they add to \( L \).

- Normalize the seasonal factors by

\[ c_t(0) = \bar{c}_t \left( \frac{L}{\sum_{t=1}^{L} c_t} \right) \quad t = 1, 2, \ldots, L \]  (1-11)

where \( c_t(0) \) is a normalized seasonal factor computed at \( T=0 \).

This procedure produces estimates \( \hat{a}_1(0) \), \( b_2(0) \) and \( \hat{c}_t(0) \) assuming that the origin of time is immediately prior to period 1. Forecasting future observations usually requires initial estimates computed with period \( mL \) as the origin of time. At that point in time the permanent component can be estimated by
\[ \hat{a}_1(\text{mL}) = \bar{x}_m + (L/2) \hat{b}_2(0) \]  

(1-12)

instead of equation (1-8) then use \( \hat{b}_2 \) and \( \hat{c}_t \) as computed previously.

The program presented in this section offers the user three options:

1. A non-smoothed forecast made at period mL, using equations (1-7), (1-12), and (1-9) to (1-11).

2. A smoothed forecast made by initially estimating the parameters \( \hat{b}_2 \), \( \hat{a}_1 \) and \( \hat{c}_t \) at period 0, using equations (1-7) to (1-11) and repeatedly smoothing them period by period according to equations (1-3) to (1-5) until values at the end of period mL are obtained. The smoothing constants A, B and G are provided by the user.

3. A smoothed forecast made by following the procedure of option 2 with the exception that A, B and G are estimated from historical data, using an error minimization procedure.
Program Documentation

Program Name: Winters

Program Function: generate a forecast of future observations based on historical data series that presents both a trend and a seasonal pattern.

Program Input: 1. number of seasons for which data is available (integer)
                2. number of periods per season (integer)
                3. past observations starting with earliest (reals)
                4. smoothing constants (optional)
                   - permanent component smoothing constant $(0 < A < 1)$
                   - trend smoothing constant $(0 < B < 1)$
                   - seasonal smoothing constant $(0 < G < 1)$

Program Logic: 1. Accept input as described above
                2. Display data and make corrections if any
                   if not go to step 2
                3. Estimate initial values of $\hat{a}_1$, $\hat{b}_2$, and $\hat{c}_t$
                   using equations (1-7), (1-12) and (1-9) to (1-11)
                4. If non-smoothing is desired, display the forecast
                5. If smoothing is desired, accept smoothing constants $(A, B, G)$ and go to step 7
6. If smoothing is desired but smoothing constants are not available, repeat up to step 7 until the set of values for A, B, and G which minimizes the error between forecasts and actual observations, is found. A, B, and G have values 0.1, 0.2, . . ., 0.9

7. Use the optimum set of values of A, B, and G and go to step 7

8. Estimate $\hat{a}_1$, $\hat{b}_2$, $c_L$ with period 0 as origin of time, using equations (1-7) to (1-11) and repeatedly smoothing season by season, until period mL is reached, then generate and display the forecast

Program Output: Forecast of future observations for the upcoming season

1. without smoothing

2. with smoothing constants provided by the user

3. with smoothing constants estimated by the program (see case example)
Program Listing

10 FOR II = 1 TO 16
20 PRINT
30 NEXT II
40 PRINT "SEASONAL FORECASTING"
50 PRINT "USING WINTER'S SMOOTHING"
60 PRINT "ITEMS DEPARTMENT"
70 PRINT "UNIVERSITY OF"
80 PRINT "CENTRAL FLORIDA"
90 FOR I=1 TO 5
100 PRINT "NEXT I"
110 FOR II=1 TO 1000
120 INPUT "ENTER THE NUMBER OF SEASONS";M
130 INPUT "ENTER THE NUMBER OF PERIODS PER SEASON";L
140 D=(M*L)-1
150 DIM A(D),B2(L),A1(L),C(D),AVG(M),X(L),SF(L),NSF(L),
160 X(D),XF(L),SO(L)
170 PRINT "ENTER OBSERVATIONS STARTING WITHEarliest";
180 FOR I=1 TO D STEP-1
190 INPUT A(I)
200 NEXT I
210 REM **DATA DISPLAY**
220 CLS
230 PRINT "DATA DISPLAY"
240 J1=D
250 FOR II=1 TO M
260 PRINT "SEASON NO.";II
270 FOR JJ=1 TO L
280 PRINT "PERIOD NO.";JJ;
290 A(J1)
300 NEXT JJ
310 NEXT II
320 INPUT "DO YOU WANT TO MAKE ANY CHANGES? YES OR NO";
330 IF A$="NO" THEN 410
340 IF A$="YES" THEN 370
350 GOTO 330
360 PRINT "ENTER SEASON NO., PERIOD NO., CHANGE"
370 INPUT S1,F1,L1
380 I=(C+1)-((S1-1)*X(L)+F1)
390 A(I)=L1
400 GOTO 230
410 REM ** COMPUTATION OF INITIAL VALUES **
420 S=D
430 FOR K=1 TO M
440 LAST=D-(K*L)+1
450 SUM=0
460 FOR J=S TO LAST STEP-1
470 SUM =SUM+A(J)
480 NEXT J
490 AUG(K)=SUM/L
510 S=D-(K*L)
520 NEXT K
530 EZ(0)=(AUG(M)-AUG(I))/(M-1)*L)
540 AI(0)=AUG(M)+(L/Z)*EZ(0)
550 S=D
560 FOR K=1 TO M
570 LAST=D-(K*L)+1
580 J=1
590 FOR I=S TO LAST STEP-1
600 C(I)=A(I)/(AUG(K)-((L+1)/2)-J)*EZ(0))
610 J=J+1
620 NEXT I
630 S=D-(K*L)
640 NEXT K
650 REM ** COMPUTE AUG. VALUES OF SEASONAL FACTORS **
660 LAST=D-L+1
670 N=D-(M-1)*L)
680 FOR I=1 TO LAST STEP-1
690 SUM=0
700 FOR J=I TO 0 STEP-L
710 SUM=SUM+C(J)
720 NEXT J
730 SF(N)=SUM/M
740 N=N-1
750 NEXT I
760 REM ** NORMALIZED SFS **
770 S=D-(M-1)*L)
780 SUM=0
790 FOR I=S TO 0 STEP-1
800 SUM=SUM+SF(I)
810 NEXT I
820 FOR J=S TO 0 STEP-1
830 NSF(J)=L*SF(J)/SUM
840 NEXT J
850 REM ** FORECAST AT PERIOD T=0 **
860 FOR I=1 TO L
870 J=L-I
880 X(I)=(AI(J)+EZ(0)*I)*NSF(J)
890 NEXT I
905 CLS
910 PRINT "DO YOU WANT TO SMOOTH THE DATA? YES OR NO" 
915 INPUT A
920 IF A="YES" THEN 925
925 CLS
930 PRINT "** FORECAST FOR SEASON NO.";(M+1);" **
935 PRINT " (WITHOUT SMOOTHING)"
PRINT "PERIOD NO. FORECAST"
FOR I=1 TO L
PRINT ";I;" ;X(I)
NEXT I
GOTO 6000
PRINT "DO YOU WANT TO PROVIDE THE SMOOTHING CONSTANTS ?
YES OR NO"
INPUT F$
IF F$="NO" THEN 2000
PRINT "PROVIDE SMOOTHING CONSTANT S"
PRINT "FOR PERMANENT COMPONENT :"
INPUT AL
PRINT "FOR TREND :"
INPUT BT
PRINT "FOR SEASONAL :"
INPUT GM
GOSUB 5000
GOSUB 6000
GOTO 8000
CLS
FOR II=1 TO 10
PRINT " THE PROGRAM IS EXECUTING"
NEXT II
FOR I=1 TO L
PRINT "THE PROGRAM IS EXECUTING"
NEXT I
K=0
FOR AL=.1 TO .9 STEP .1
FOR BT=.1 TO .9 STEP .1
FOR GM=.1 TO .9 STEP .1
K=K+1
SUM=0
GOSUB 5000
TEST=SUM
IF K=1 THEN 3010
IF MIN>TEST THEN 3010
GOTO 3050
MIN=TEST:AC=AL:BO=BT:GO=GM
LA=A11:LB=B22
FOR J=6 TO 0 STEP -1
SC(J)=NSF(J)
NEXT J
NEXT GM
THE OPTIMUM SET OF SMOOTHING CONSTANTS, IS:

PERMANENT COMPONENT = \$AO
TREND COMPONENT = \$EO
SEASONAL COMPONENT = \$GO

FOR II=1 TO 9000
NEXT II
FOR I=S TO 0 STEP -1
NSF(I)=NSO(I)
NEXT I
PRINT "TO DISPLAY THE FORECAST, TYPE D"
INPUT D$=
IF D$="D" THEN 4065
GOTO 6000
GOSUE 6000
PRINT " "
RETURN

REM THIS SUBROUTINE UPDATES THE SMOOTHING CONSTANTS
A1(0)=AVG(1)-((L/2)*EC(0))
J=S
FOR I=0 TO 0 STEP -1
IF J=-1 THEN J=S
A11=(AL*(A(I)/NSF(J)))+((1-AL)*(A1(0)+EC(0)))
B22=(BT*(A11-A1(0)))+((1-BT)*EC(0))
X1=(GM*(A(I)/A11))+(1-GM)*NSF(J)
ERS=(X1-A(I))*X2
SUM=SUM+ERS
NSF(J)=X1
A1(0)=A11
B2(0)=B22
J=J-1
NEXT I
RETURN
REM THIS SUBROUTINE PRINTS THE FORECAST
FORECAST FOR SEASON NO.;(M+1); "** (WITH SMOOTHING)"

6025 PRINT "        PERIOD NO.    FORECAST"
6030 J=0
6040 FOR I=S TO 0 STEP -1
6050 J=J+1
6060 XF=(A11+(B22*J))*NSF(I)
6070 PRINT "     ";S-I+1;"     ";XF
6080 NEXT I
6090 RETURN
8000 PRINT "DO YOU WANT ANOTHER RUN? YES OR NO"
8005 INPUT B$"
8010 IF B$="YES" THEN 225
8020 END
Case Example

An analyst is faced with the problem of forecasting the demand for a product for each quarter of the upcoming year (1980). Data for the past six years (1974 to 1979) is available. Table 1-1 shows the data collection.

TABLE 1-1

SAMPLE DATA

<table>
<thead>
<tr>
<th>Year</th>
<th>Period</th>
<th>Sales ($)</th>
<th>Year</th>
<th>Period</th>
<th>Sales ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1974</td>
<td>Fall</td>
<td>362</td>
<td>1977</td>
<td>Fall</td>
<td>544</td>
</tr>
<tr>
<td></td>
<td>Winter</td>
<td>385</td>
<td></td>
<td>Winter</td>
<td>582</td>
</tr>
<tr>
<td></td>
<td>Spring</td>
<td>432</td>
<td></td>
<td>Spring</td>
<td>681</td>
</tr>
<tr>
<td></td>
<td>Summer</td>
<td>341</td>
<td></td>
<td>Summer</td>
<td>557</td>
</tr>
<tr>
<td>1975</td>
<td>Fall</td>
<td>382</td>
<td>1978</td>
<td>Fall</td>
<td>628</td>
</tr>
<tr>
<td></td>
<td>Winter</td>
<td>409</td>
<td></td>
<td>Winter</td>
<td>707</td>
</tr>
<tr>
<td></td>
<td>Spring</td>
<td>498</td>
<td></td>
<td>Spring</td>
<td>773</td>
</tr>
<tr>
<td></td>
<td>Summer</td>
<td>387</td>
<td></td>
<td>Summer</td>
<td>592</td>
</tr>
<tr>
<td>1976</td>
<td>Fall</td>
<td>473</td>
<td>1979</td>
<td>Fall</td>
<td>627</td>
</tr>
<tr>
<td></td>
<td>Winter</td>
<td>513</td>
<td></td>
<td>Winter</td>
<td>725</td>
</tr>
<tr>
<td></td>
<td>Spring</td>
<td>582</td>
<td></td>
<td>Spring</td>
<td>854</td>
</tr>
<tr>
<td></td>
<td>Summer</td>
<td>474</td>
<td></td>
<td>Summer</td>
<td>661</td>
</tr>
</tbody>
</table>


Notice that the data presents a seasonal pattern. For example, the Fall sales are usually the lowest among the four seasons, and the Spring sales the highest. Another observation is that there is a continuous increase in sales from year to year. i.e., the data has a positive trend.

[The user can load the program either from tape or mini-disk. Execution is started by typing RUN. The screen will display]
The user will then be asked to input the data pertaining to his problem. For our case:

ENTER THE NUMBER OF SEASONS ? 6

ENTER THE NUMBER OF PERIODS PER SEASON ? 4
ENTER OBSERVATIONS STARTING WITH THE EARLIEST

?362
?385
?432
?391
?382
?409
?498
?387
?473
?513
?582
?4754
?544
?582
?681
?557
?628
?707
?773
?592

The program displays the data entered and gives the user the opportunity to correct it. This is done by responding YES to the question "DO YOU WANT TO MAKE ANY CHANGES? YES OR NO?" and entering the
corrected data with the format SEASON NO., PERIOD NO., CHANGE.

DO YOU WANT TO MAKE ANY CHANGES? YES OR NO  
ENTER SEASON NO., PERIOD NO., CHANGE  
?3,4,474

DATA DISPLAY

<table>
<thead>
<tr>
<th>SEASON NO.</th>
<th>PERIOD NO. 1</th>
<th>PERIOD NO. 2</th>
<th>PERIOD NO. 3</th>
<th>PERIOD NO. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>362</td>
<td>385</td>
<td>432</td>
<td>341</td>
</tr>
<tr>
<td>2</td>
<td>382</td>
<td>409</td>
<td>498</td>
<td>387</td>
</tr>
<tr>
<td>3</td>
<td>473</td>
<td>513</td>
<td>582</td>
<td>474</td>
</tr>
<tr>
<td>4</td>
<td>544</td>
<td>582</td>
<td>681</td>
<td>597</td>
</tr>
<tr>
<td>5</td>
<td>628</td>
<td>707</td>
<td>773</td>
<td>592</td>
</tr>
<tr>
<td>6</td>
<td>627</td>
<td>725</td>
<td>854</td>
<td>861</td>
</tr>
</tbody>
</table>

Corrections are made one at a time, and the data is displayed after each correction.
<table>
<thead>
<tr>
<th>Season No. 1</th>
<th>Period No. 1</th>
<th>362</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Period No. 2</td>
<td>385</td>
</tr>
<tr>
<td></td>
<td>Period No. 3</td>
<td>432</td>
</tr>
<tr>
<td></td>
<td>Period No. 4</td>
<td>341</td>
</tr>
<tr>
<td>Season No. 2</td>
<td>Period No. 1</td>
<td>382</td>
</tr>
<tr>
<td></td>
<td>Period No. 2</td>
<td>407</td>
</tr>
<tr>
<td></td>
<td>Period No. 3</td>
<td>498</td>
</tr>
<tr>
<td></td>
<td>Period No. 4</td>
<td>387</td>
</tr>
<tr>
<td>Season No. 3</td>
<td>Period No. 1</td>
<td>473</td>
</tr>
<tr>
<td></td>
<td>Period No. 2</td>
<td>513</td>
</tr>
<tr>
<td></td>
<td>Period No. 3</td>
<td>582</td>
</tr>
<tr>
<td></td>
<td>Period No. 4</td>
<td>474</td>
</tr>
<tr>
<td>Season No. 4</td>
<td>Period No. 1</td>
<td>544</td>
</tr>
<tr>
<td></td>
<td>Period No. 2</td>
<td>582</td>
</tr>
<tr>
<td></td>
<td>Period No. 3</td>
<td>681</td>
</tr>
<tr>
<td></td>
<td>Period No. 4</td>
<td>557</td>
</tr>
<tr>
<td>Season No. 5</td>
<td>Period No. 1</td>
<td>628</td>
</tr>
<tr>
<td></td>
<td>Period No. 2</td>
<td>707</td>
</tr>
<tr>
<td></td>
<td>Period No. 3</td>
<td>773</td>
</tr>
<tr>
<td></td>
<td>Period No. 4</td>
<td>592</td>
</tr>
<tr>
<td>Season No. 6</td>
<td>Period No. 1</td>
<td>627</td>
</tr>
<tr>
<td></td>
<td>Period No. 2</td>
<td>725</td>
</tr>
<tr>
<td></td>
<td>Period No. 3</td>
<td>854</td>
</tr>
<tr>
<td></td>
<td>Period No. 4</td>
<td>661</td>
</tr>
</tbody>
</table>

Do you want to make any changes? Yes or no. No.

Do you want to smooth the data? Yes or no. No.

*User's response to the question.

At this point the program generates a forecast without smoothing and offers the user the option to display it.
If the user chooses to smooth the data instead, and can provide the smoothing constants, he will answer the corresponding question by YES.

DO YOU WANT TO MAKE ANY CHANGES? YES OR NO NO
DO YOU WANT TO SMOOTH THE DATA ? YES OR NO YES
DO YOU WANT TO PROVIDE THE SMOOTHING CONSTANTS ? YES

PROVIDE SMOOTHING CONSTANTS
FOR PERMANENT COMPONENT = .2
FOR TREND = .1
FOR SEASONAL = .05

** FORECAST FOR SEASON NO.7 **
(WITH SMOOTHING)

<table>
<thead>
<tr>
<th>PERIOD NO.</th>
<th>FORECAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>744.70266</td>
</tr>
<tr>
<td>2</td>
<td>806.629234</td>
</tr>
<tr>
<td>3</td>
<td>919.591437</td>
</tr>
<tr>
<td>4</td>
<td>719.123868</td>
</tr>
</tbody>
</table>
DO YOU WANT ANOTHER RUN? YES OR NO YES

If the user chooses to smooth the data, but cannot provide the smoothing constants:

DO YOU WANT TO MAKE ANY CHANGES? YES OR NO NO
DO YOU WANT TO SMOOTH THE DATA? YES OR NO YES
DO YOU WANT TO PROVIDE THE SMOOTHING CONSTANTS? YES OR NO NO

While it is estimating the set of values of the smoothing constants that best fits the historical data, the program will display:

THE PROGRAM IS EXECUTING
PLEASE STAND BY!!

After a few minutes, these values are displayed and the forecast is displayed after typing D.

THE OPTIMUM SET OF SMOOTHING CONSTANTS COMPUTED FROM HISTORICAL DATA IS:

PERMANENT COMPONENT = .4
TREND COMPONENT = .1
SEASONAL COMPONENT = .3

TO DISPLAY THE FORECAST TYPE D THEN HIT RETURN TO DISPLAY THE FORECAST TYPE D THEN HIT RETURN D
** FORECAST FOR SEASON NO. 7 **
(WITH SMOOTHING)

<table>
<thead>
<tr>
<th>PERIOD NO.</th>
<th>FORECAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>735.9347</td>
</tr>
<tr>
<td>2</td>
<td>809.613724</td>
</tr>
<tr>
<td>3</td>
<td>922.995227</td>
</tr>
<tr>
<td>4</td>
<td>719.193995</td>
</tr>
</tbody>
</table>

In all cases, after the forecast is displayed, the user is asked:

**DO YOU WANT ANOTHER RUN, YES OR NO**

If his answer is YES, the data entered previously is used again and the opportunity to modify it is given. If his answer is NO, the program execution stops.

If monthly, rather than quarterly, data was available the procedure would be the same, except that the number of periods per season would be 12. The program would then output 12 monthly forecasts.
CHAPTER II

THE LEARNING CURVE

Introduction

Industrial engineers, human engineers and other professionals have recognized that learning is time dependent. The Learning Curve is an accepted tool of Management for determining manpower requirements, estimating the costs of future production and controlling shop labor where performance time decreases as a result of learning.

It is the purpose of this section to present a computer program written in Interactive Basic, which uses the Learning Curve (Jordan 1966) to provide the user with decision tools pertaining to production, man-power and costs.

Theory

The theory of the Learning Curve proposes that as the total quantity of units produced doubles, the time per unit declines at some constant percentage.

For example, if it is expected that a 90 percent rate of improvement (or learning efficiency) will be experienced, then, as production doubles, the average time per unit will decline 10 percent. Figure 2-1 is an example of a chart used to follow the
progress of an employee through the first 26 units produced on a 90% curve, using 100 as the time to produce the first unit (Jordan 1966).

![Progress Chart at 90% learning efficiency](image)

Fig. 2-1. Progress Chart at 90% learning efficiency

Observe that the smaller the percent rate of improvement, the greater the progressive improvement with production output. Typical rates of learning that have been encountered are:

- 70 - 80% for large or fine assembly work (such as aircraft)
80 - 90% for welding
90 - 95% for machinery

The Learning Curve is a hyperbola of the form:

\[ Y_X = K X^N \]  \hspace{1cm} (2-1)

where

\( Y_X \) = cumulative average time per unit for \( X \) units produced
\( K \) = time to produce the first unit (constant)
\( X \) = number of units (cycles) produced
\( N \) = exponent representing the hyperbolic relationship

By definition, the learning efficiency, in percent, is equal to

\[ \frac{Y_{2X}}{Y_X} = \frac{K(2X)^N}{K(X)^N} = 2^N \]  \hspace{1cm} (2-2)

And taking the log of both sides

\[ N = \frac{\log \text{ of learning efficiency}}{\log 2} \]  \hspace{1cm} (2-3)

Once \( N \) has been determined, equation (2-1) can be used to find values of \( Y_X \) given \( X \) and values of \( X \) given \( Y_X \). Cost and manpower estimates can then be deduced from \( X \) and \( Y_X \).
Program Documentation

Program Name: Learning Curve

Program Function: provide estimates for performance times, labor costs and production volumes as functions of the learning efficiency

Program Input:
1. number of observations to be entered (integer)
2. actual observations, with format: unit number, man-hrs. to produce it

Program Logic:
1. accept input
2. display data, make corrections if any, if not, go to step 2
3. compute cumulative averages
4. derive the exponent N from equation (2-3)
5. display the modes of operation (see case example)
6. if mode 1 is selected, the learning efficiency is displayed
7. if mode 2 is selected, the total number of man-hours to produce X units is derived from equation (2-1)
8. if mode 3 is selected, the production level X at an average man-hrs./unit of Yx is derived from equation (2-1)
9. if mode 4 is selected, the labor cost of producing X units is computed
10. if mode 5 is selected, program execution is resumed

Program Output*:  
- mode 1: displays the learning efficiency
- mode 2: displays the total man-hrs. to produce X units
- mode 3: displays the number of units X that could be produced at an average rate of \( Y_x \) man-hrs./units
- mode 4: displays the labor cost of producing X units

(See case example)

* Self explanatory messages are displayed when the wrong data type or sequence is used.
Program Listing

5 REM GREETING PROGRAM
10 FOR II=1 TO 16
15 PRINT " "
20 NEXT II
25 PRINT " LEARNING CURVE"
30 PRINT " ITEMS DEPARTMENT"
35 PRINT " UNIVERSITY OF"
40 PRINT " CENTRAL FLORIDA"
45 PRINT " 
50 FOR I=1 TO 5
55 PRINT " 
60 NEXT I
65 FOR II=1 TO 1000
70 NEXT II
72 GOSUB 9000
74 PRINT " TO START DATA ENTRY TYPE S THEN RETURN "
76 PRINT " TO EXIT THE PROGRAM TYPE E THEN RETURN "
78 INPUT A$ 27
80 IF A$="S" THEN 100
82 IF A$="E" THEN 9000
84 GOTO 78
100 PRINT " HOW MANY OBSERVATIONS DO YOU WANT TO ENTER ?"
101 INPUT L
102 PRINT " 
105 DIM M(L),AVG(L)
110 PRINT " ENTER UNIT NO.,MAN-HRS ? "
120 FOR I=1 TO L
125 INPUT U,M(I)
130 IF U=I THEN 170
135 PRINT " 
140 PRINT " UNIT NO. IS OUT OF SEQUENCE"
145 PRINT " REENTER PROPER UNIT NO.,MAN-HRS"
150 GOTO 130
155 NEXT I
160 REM DATA DISPLAY
165 PRINT " "
170 PRINT " UNIT NO. MAN-HRS"
174 PRINT " "
175 FOR I=1 TO L
176 PRINT " ;I:"
177 PRINT " ;M(I)"
178 NEXT I
180 PRINT " 
181 PRINT
182 PRINT " DO YOU WANT TO MAKE ANY CHANGES ,YES OR NO ?"
183 INPUT A$ 27
184 IF A$="NO" THEN 290
186 IF A$="YES" THEN 190
188 GOTO 182
189 PRINT
190 PRINT
192 PRINT " ENTER UNIT NO.,MAN-HRS"
196 ON ERROR GOTO 205
200 INPUT I,M(I)
204 GOTO 172
205 PRINT
206 PRINT "** ERROR IN DATA ENTRY ** :"
207 PRINT "UNIT NO. EXCEEDS THE ";L;" OBSERVATIONS LIMIT"
208 PRINT
209 GOTO 190
290 REM COMPUTE LEARNING CURVE PERCENTAGE
300 SUM=0
310 FOR I=1 TO L
320 SUM=SUM+M(I)
330 AVG(I)=SUM/I
340 NEXT I
350 K=0
360 ADD=0
370 FOR I=1 TO L
380 J=2*I
390 IF J>L THEN 450
400 EFF=AVG(J)/AVG(I)
410 K=K+1
420 ADD=ADD+EFF
430 I=J
440 GOTO 360
450 NEXT I
460 PER = ADD/K
470 N=LOG(PER)/LOG(2)
480 CST=M(I)
490 GOSUB 6000
500 PRINT "ENTER SELECTED OPTION :"
505 INPUT O
510 IF O=1 THEN 600
520 IF O=2 THEN 800
530 IF O=3 THEN 1000
540 IF O=4 THEN 1200
550 IF O=5 THEN 1400
560 GOTO 500
600 GOSUB 700
610 GOTO 490
700 REM DISPLAY LEARNING EFFICIENCY
705 CLS
710 FOR II=1 TO 10
715 PRINT
720 NEXT II
725 PCT=PER*100
730 PRINT "LEARNING EFFICIENCY = ";PCT;"%"
735 PRINT
740 PRINT "TYPE C TO CONTINUE PROGRAM EXECUTION : "
742 INPUT C$
745 IF C$="C" THEN 790
750 GOTO 740
790 RETURN
800 GOSUB 900
810 GOTO 490
900 REM FIND MAN-HRS GIVEN X
905 CLS
910 PRINT
915 PRINT "ENTER THE NO. OF UNITS TO BE PRODUCED:"
917 INPUT X
920 Y=CST*(X^N)
922 PRINT
925 PRINT "PRODUCTION = ";X;" UNITS"
930 PRINT "AVG. MAN-HRS/UNIT = ";Y
935 PRINT
940 PRINT
950 PRINT "DO YOU WANT TO SELECT THIS OPTION AGAIN, YES OR NO?"
955 INPUT O$
960 IF O$="YES" THEN 910
970 RETURN
1000 GOSUB 1100
1010 GOTO 490
1100 REM FIND PRODUCTION LEVEL
1105 CLS
1110 PRINT
1115 PRINT "ENTER THE AVG. MAN-HRS/UNIT:"
1117 INPUT Y
1120 X=(Y/CST)*(1/N)
1125 PRINT
1130 PRINT "AVG. MAN-HRS/UNIT = ";Y
1135 PRINT "PRODUCTION = ";X;" UNITS"
1140 PRINT
1145 PRINT
1150 PRINT "DO YOU WANT TO SELECT THIS OPTION AGAIN, YES OR NO?"
1155 INPUT A$
1158 IF A$="YES" THEN 1110
1160 RETURN
1200 GOSUB 1300
1210 GOTO 490
1300 REM FIND LABOR COST
1305 CLS
1307 PRINT
1310 PRINT "ENTER QUANTITY TO PRODUCE, DOLLARS/MAN-HR?"
1312 INPUT X,C
1315 Y=CST*(X^N)
1320 T=Y*X*C
1325 PRINT
1330 PRINT "LABOR COST = ";T
1335 PRINT
3040 PRINT
3045 PRINT "DO YOU WANT TO SELECT THE SAME OPTION AGAIN,
YES OR NO?"
3047 INPUT A$
3050 IF A$="YES" THEN 1307
3055 RETURN
1340 GOTO 9000
3000 REM HEADER DISPLAY
3005 CLS
3010 PRINT "" THIS PROGRAM IS BASED ON THE"
3015 PRINT " LEARNING CURVE THEORY."
3020 PRINT " IT ACCEPTS A SEQUENCE OF AT"
3025 PRINT " LEAST TWO OBSERVATIONS."
3030 PRINT " EACH OBSERVATION HAS THE"
3035 PRINT " FOLLOWING FORMAT:"
3040 PRINT " UNIT NO. * MAN-HRS TO PRODUCE IT"
3045 PRINT " ** PRESS RETURN AFTER EACH"
3050 PRINT " OBSERVATION"
3060 PRINT
3065 RETURN
6000 REM DISPLAY OF OPTION
6005 CLS
6010 PRINT "" SELECT FROM THE FOLLOWING OPTIONS:"
6015 PRINT " 1. FIND LEARNING CURVE EFFICIENCY"
6020 PRINT " 2. FIND AVG. MAN-HRS/UNIT GIVEN"
6025 PRINT " A PRODUCTION OF X UNITS"
6030 PRINT " 3. FIND THE PRODUCTION LEVEL"
6035 PRINT " GIVEN THE AVG. MAN-HRS/UNIT"
6040 PRINT " 4. FIND THE LABOR COST OF"
6045 PRINT " PRODUCING X UNITS"
6050 PRINT " 5. STOP PROGRAM EXECUTION"
6055 RETURN
9000 END
Case Example

After loading the program and typing RUN, the screen displays

LEARNING CURVE

ITEMS DEPARTMENT
UNIVERSITY OF
CENTRAL FLORIDA

followed by a brief statement of the program function

THIS PROGRAM IS BASED ON THE
LEARNING CURVE THEORY.
IT ACCEPTS A SEQUENCE OF AT
LEAST TWO OBSERVATIONS,
EACH OBSERVATION HAS THE
FOLLOWING FORMAT:
UNIT NO., MAN-HRS TO PRODUCE IT
** PRESS RETURN AFTER EACH OBSERVATION

This case example involves the manufacturing of a product with the following performance times

<table>
<thead>
<tr>
<th>UNIT NO.</th>
<th>MAN-HRS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>66</td>
</tr>
</tbody>
</table>

The program is started by typing S
TO START DATA ENTRY TYPE S THEN RETURN
TO EXIT THE PROGRAM TYPE E THEN RETURN
S
HOW MANY OBSERVATIONS DO YOU WANT TO ENTER?
4

ENTER UNIT NO., MAN-HRS?
1  100
2  80
3  75
4  70

UNIT NO.    MAN-HRS
1         100
2          80
3          75
4          70

DO YOU WANT TO MAKE ANY CHANGES YES OR NO?
YES

corrections are made with the format UNIT NO., MAN-HRS. and the
corrected data displayed

ENTER UNIT NO., MAN-HRS
4  66

UNIT NO.    MAN-HRS
1         100
2          80
3          75
4        66

DO YOU WANT TO MAKE ANY CHANGES YES OR NO?
NO
The program options are then displayed

SELECT FROM THE FOLLOWING OPTIONS :

1. FIND LEARNING CURVE EFFICIENCY  
2. FIND AVG. MAN-HRS/UNIT GIVEN A PRODUCTION OF X UNITS  
3. FIND THE PRODUCTION LEVEL GIVEN THE AVG. MAN-HRS/UNIT  
4. FIND THE LABOR COST OF PRODUCING X UNITS  
5. STOP PROGRAM EXECUTION

Sample outputs for different option selections are

ENTER SELECTED OPTION :
1

LEARNING EFFICIENCY = 89.5833 %

TYPE C TO CONTINUE PROGRAM EXECUTION :
C

ENTER SELECTED OPTION :
2

ENTER THE NO. OF UNITS TO BE PRODUCED :
200

PRODUCTION = 200 UNITS  
AVG. MAN-HRS/UNIT = 43.1352

DO YOU WANT TO SELECT THIS OPTION AGAIN, YES OR NO
NO
ENTER SELECTED OPTION:
3

ENTER THE AVG. MAN-HRS/UNIT
45

AVG. MAN-HRS/UNIT = 45
PRODUCTION = 153.183 UNITS

DO YOU WANT TO SELECT THIS OPTION AGAIN, YES OR NO
NO

ENTER SELECTED OPTION:
4

ENTER QUANTITY TO PRODUCE, DOLLARS/MAN-HR?
300

LABOR COST = $ $ 72804.7

DO YOU WANT TO SELECT THE SAME OPTION AGAIN, YES OR NO
NO

ENTER SELECTED OPTION:
5

The program options are displayed after each selection.
CHAPTER III

LINE OF BALANCE (LOB)

Introduction

Line of Balance (LOB) is a technique for production control. Its best application is in situations where parts and subassemblies make up the final product. LOB can be used to monitor and control the progress during the production cycle. The technique identifies bottlenecks and delays in production schedule, so that corrective measures can be taken in order to meet delivery deadlines.

The theory of LOB is used here in a computer program to present an alternative to the traditional graphical procedure.

Theory

The Line of Balance technique was developed as a graphic method of industrial programming using the contractual requirements against achievements.

LOB relates actual status of the elements of a production program to planned progress. It identifies those elements which are lagging prior to delay in delivery of the end item. It provides an indication as to how well the various phases of manufacturing are synchronized.
Line-of-Balance Charts

The basic unit of the LOB system is the production plan. Figure 3-1 shows an example of such a plan. The time units are shown to scale and are measured in days prior to shipment point. At points 1 and 6, purchased parts must be made available. At points 2 and 4, parts from company-made stock must be introduced. At point 3, a purchased but fabricated part finished under subcontract must be available. At point 5, parts 2 and 3 are combined; the period between points 5 and 9 is the assembly time required to prepare the resulting subassembly B. Parts 1 and 6 are combined with company-made part 4 at point 7. The assembly time to prepare this subassembly A is the time between points 7 and 8. Subassemblies A and B are combined between points 10 and 11 into the final assembly. Between points 11 and 12, two days are required for the packaging and shipping process.

The objective chart and the progress chart make up the remaining two portions of the LOB package. The objective chart is a
cumulative calendar schedule of number of production units to be shipped. The progress chart is drawn to the same vertical scale as the objective chart and is placed on a common ordinate axis. Figure 3-2 shows both charts side to side for the production cycle shown in Figure 3-1. On the objective chart, the actual delivery of units is plotted below the contract schedule. On the progress chart, the vertical bars represent the actual availability of components and subassemblies. The procedure for striking the line of balance is as follows:

1. Starting with the study data on the horizontal axis of the objective chart, mark off the lead time for each control point on the production plan.
2. Project up to the cumulative contract schedule curve from each point identified in step 1.
3. Project the points identified on the cumulative contract schedule horizontally to the progress chart.
4. Plot the line of balance.

Fig. 3-2. Drawing the LOB
Figure 3.2 demonstrates this procedure for control points 2 and 3. There is a 22 working days (1 month) lead time associated with these points as shown in Figure 3.1. Projecting, we find that the line of balance is struck at 52 units. Thus, if we expect to meet our production objective, 52 units should have passed control points 2 and 3. Actually, only 45 have passed control point 2 and 49 control point 3. All bars which fall below the line of balance represent points which are behind the production plan; those above the line of balance indicate points which are ahead of schedule.

Review of Figure 3.2 indicates that part 7 also depends on parts 1, 4, and 6. Parts 1 and 6 are in oversupply, and part 4 is substantially behind in actual completion, but nearing completion for 90 percent. Accordingly, there may be an opportunity to catch up rapidly on subassembly A production. Item 8 is the completion of subassembly A and is a function of item 7. Items 10, 11, and 12 are also below production requirements, but these are also a function of subassembly A. Accordingly, the line of balance for this progress report would indicate that the efforts should be increased for company-made part 4 so that subassembly A production can be increased. Overtime might be justified in this case, or it might be necessary to slow down other lines to benefit subassembly A.
Program Documentation

Program Name: Line of Balance

Program Function: Identify those phases, in a manufacturing cycle, which are behind schedule, prior to delay in delivery of the end product.

Program Input:

1. the number of key stations (control points) in the production cycle (integer)
2. the number of scheduled deliveries (integer)
3. time to produce one complete unit of the product (in working days) note: 22 working days per month
4. delivery schedule with formal: DAY, MONTH, YEAR, QUANTITY TO DELIVER
5. lead time, present status (for each control point) lead time = time interval between a control point and total completion of the product
6. production start date: DAY, MONTH, YEAR
7. study date DAY, MONTH, YEAR

Program Logic:

1. accept input
2. display data and make corrections if any, if not go to step 2
3. sort delivery dates
4. compute cumulative delivery quantities
5. repeat steps 5 to 7 for each control point
6. add lead time to study date to generate the control point study date
7. find the delivery points surrounding the control point study date
8. interpolate between these 2 dates and their cumulative quantities to find the line of balance for the control point
9. display output

Program Output: - CONTROL POINT, PRESENT STATUS, LOB, DEVIATION

An asterisk (*) preceeds those control points which are behind schedule (See case example).
Program Listing

5 FOR II=1 TO 16
10 PRINT "LINE OF BALANCE"
15 NEXT II
20 PRINT "IEMS DEPARTMENT"
25 PRINT "UNIVERSITY OF"
30 PRINT "CENTRAL FLORIDA"
35 PRINT "FOR I=1 TO 5
40 PRINT "NEXT I"
45 PRINT "FOR II=1 TO 1000
50 NEXT II
55 FOR II=1 TO 1000
60 NEXT II
65 CLS
70 PRINT "THIS PROGRAM IS BASED ON THE LINE OF
71 PRINT "BALANCE THEORY. ITS OBJECTIVE IS TO
75 PRINT "COMPARE THE ACTUAL STATUS OF KEY OPERATIONS
80 PRINT "(CONTROL POINTS) IN A MANUFACTURING CYCLE,"
85 PRINT "TO A CONTRACTUAL DELIVERY SCHEDULE."
90 PRINT "THE INPUT TO THE PROGRAM CONSISTS OF :
95 PRINT "- THE DELIVERY SCHEDULE."
100 PRINT "- LEAD TIME & PROGRESS (AT EACH CONTROL POINT)"
105 PRINT "- DATE AT WHICH THE STUDY IS MADE"
110 PRINT "LEAD TIME: TIME INTERVAL BETWEEN A CONTROL
115 PRINT "POINT AND TOTAL COMPLETION OF THE PRODUCT. MEASURED IN
120 PRINT "WORKING DAYS."
125 PRINT "INPUT "PRESS S TO START DATA INPUT ";S$"
130 IF S$="S" THEN 205
135 INPUT "PRESS S TO START DATA INPUT ";S$"
140 IF S$="S" THEN 205
145 GOTO 135
200 REM ** DATA INPUT **
205 CLS
210 INPUT "ENTER NO. OF CONTROL POINTS "; L
215 PRINT ""
220 INPUT "ENTER NO. OF SCHEDULED DELIVERIES ";S
225 PRINT ""
230 INPUT "ENTER TIME TO PRODUCE ONE UNIT (WORKING DAYS)
235 PRINT ""
240 DIM D(S),M(S),Y(S),G(S),LD(L),P(L),E(L),ND(L),NM(L),
245ED(L),EM(L),EY(L),SQ(L),EQ(L),LCB(L),DEV(L)
250 PRINT ""
255 PRINT "ENTER DELIVERY SCHEDULE AS FOLLOWS :
260 PRINT "DAY, MONTH, YEAR, QUANTITY TO BE DELIVERED"
265 PRINT "NOTE: 22 WORKING DAYS PER MONTH"
270 FOR I=1 TO S
290 INPUT D,H,Y,O
300 IF D>22 OR D<1 THEN 340
310 IF M>12 OR M<1 THEN 360
320 D(I)=D;M(I)=M;Y(I)=Y;Q(I)=Q
330 GOTO 360
340 PRINT "** INPUT ERROR : DAYS EXCEED 22 **
350 GOTO 290
360 PRINT "** INPUT ERROR : MONTHS EXCEED 12 **
370 GOTO 290
380 NEXT I
390 REM ** DATA DISPLAY **
400 CLS
410 PRINT "DELIVERY SCHEDULE"
420 PRINT "DATE QUANTITY"
430 FOR I=1 TO S
440 PRINT ";D(I);";";M(I);";";Y(I);";";G(I)
450 NEXT I
460 PRINT "DO YOU WANT TO MAKE ANY CHANGES ? YES OR NO ";A$
470 IF A$="NC" THEN 560
480 IF A$="YES" THEN 520
490 GOTO 480
500 PRINT "ENTER DELIVERY POINT, DAY, MONTH, YEAR, QUANTITY"
510 INPUT I,C1,C2,C3,C4
520 D(I)=C1;M(I)=C2;Y(I)=C3;Q(I)=C4
530 GOTO 400
540 REM ** PROCESS FLOW & PROGRESS INPUT **
550 CLS
560 PRINT "FOR EACH CONTROL POINT, ENTER :
570 PRINT "LEAD TIME, PRESENT STATUS"
580 FOR I=1 TO L
590 INPUT E,P
600 IF E<0 OR E>T THEN 610
610 PRINT "** INPUT ERROR : LEAD TIME EXCEEDS UNIT PRODUCTION TIME **
620 GOTO 590
630 NEXT I
640 REM ** LEAD TIMES & PROGRESS DISPLAY **
650 CLS
660 PRINT "CONTROL POINT LEAD TIME PROGRESS"
670 FOR I=1 TO L
680 PRINT TAB(5);I;TAB(29);LD(I);TAB(48);P(I)
690 NEXT I
700 PRINT "DO YOU WANT TO MAKE ANY CHANGES? YES OR NO ";A$
710 IF A$="NO" THEN 870
720 IF A$="YES" THEN 830
820 GOTO 790
830 PRINT "ENTER CONTROL POINT, LEAD TIME, PROGRESS"
840 INPUT I, C2, C3
850 LD(I) = C2: P(I) = C3
860 GOTO 700
870 REM ** PRODUCTION START & STUDY DATE INPUT **
880 INPUT "ENTER PRODUCTION START DATE(DAY, MTH, YR)"; PRD, 
             FRM, FRY
890 INPUT "ENTER STUDY DATE(DAY, MTH, YR)"; SD, SM, SY
900 REM ** SORT DELIVERY DATES **
910 SS = S - 1
920 FOR I = 1 TO SS
930 J = I + 1
940 IF Y(J) > Y(I) THEN 1160
950 IF Y(J) = Y(I) THEN 1080
960 GOTO 1120
970 IF M(J) > M(I) THEN 1160
980 IF M(J) = M(I) THEN 1110
990 GOTO 1120
1000 IF D(J) > D(I) THEN 1160
1010 TEMP = D(I): D(J) = D(J): D(I) = TEMP
1020 TEMP = M(I): M(J) = M(J): M(I) = TEMP
1030 TEMP = Y(I): Y(J) = Y(J): Y(I) = TEMP
1040 FLAG = 1
1050 NEXT I
1060 IF FLAG > 0 GOTO 1000
1070 REM ** DISPLAY OF OBJ. CHART **
1080 SUM = 0
1090 FOR I = 1 TO S
1100 Q(I) = Q(I) + SUM
1110 SUM = Q(I)
1120 NEXT I
1130 CLS
1140 PRINT "OBJECTIVE CHART"
1150 PRINT "CUMULATIVE"
1160 PRINT "QUANTITY"
1170 PRINT "SORTED DELIVERIES"
1180 FOR I = 1 TO S
1190 PRINT D(I); "/"; M(I); "/"; Y(I); "; Q(I)"
1200 NEXT I
1210 REM ** SRIKING THE LCB **
1220 FOR I = 1 TO L
1230 ND(I) = SD + LD(I)
1240 IF ND(I) <= 22 THEN 2120
1250 R = ND(I) / 22
1260 ND(I) = INT((R - INT(R)) * 22)
1270 NM(I) = SM + INT(R)
1280 IF NM(I) <= 12 THEN 2140
1290 R = NM(I) / 12
1300 NY(I) = SY + INT(R)
1310 NEXT I
2110 GOTO 2160
2120 NM(I)=SM:NY(I)=SY
2130 GOTO 2160
2140 NY(I)=SY
2150 GOTO 2160
2160 J=1
2170 IF NY(I)>Y(J) THEN 2250
2180 IF NY(I)=Y(J) THEN 2280
2190 IF J=1 THEN 2310
2200 JJ=J-1
2210 SD(I)=D(JJ):SM(I)=M(JJ):SY(I)=Y(JJ)
2220 ED(I)=D(JJ):EM(I)=M(J):YE(I)=Y(J)
2230 SQ(I)=Q(JJ):EQ(I)=Q(J)
2240 GOTO 2430
2250 J=J+1
2260 IF J>S THEN 2410
2270 GOTO 2170
2280 IF NM(I)>M(J) THEN 2350
2290 IF NM(I)=M(J) THEN 2380
2300 GOTO 2190
2310 SD(I)=D(JJ):SM(I)=M(JJ):SY(I)=Y(JJ)
2320 ED(I)=D(JJ):EM(I)=M(J):YE(I)=Y(J)
2330 SQ(I)=Q(JJ):EQ(I)=Q(J)
2340 GOTO 2430
2350 J=J+1
2360 IF J>S THEN 2410
2370 IF NY(I)<Y(J) THEN 2200
2375 GOTO 2290
2380 IF ND(I)>D(J) THEN 2250
2390 IF ND(I)=D(J) THEN 2280
2400 GOTO 2190
2410 PRINT "STUDY DATE EXCEEDS LAST DELIVERY DATE"
2420 GOTO 2470
2430 IF EY(I)>SY(I) THEN 2470
2440 IF EY(I)=SY(I) THEN 2470
2450 PRINT "ERROR IN SORTING DELIVERY DATES"
2460 GOTO 2800
2470 X=((EY(I)-SY(I)-1)*22+12)+(12-SM(I))*22)
2480 GOTO 2510
2490 X=((EM(I)-SM(I)-1)*22)+(22-SD(I))+ED(I)
2500 GOTO 2510
2510 IF NY(I)>SY(I) THEN 2560
2520 IF NY(I)=SY(I) THEN 2560
2530 PRINT "ERROR IN SORTING DELIVERY DATES"
2540 GOTO 2760
2550 X1=((NY(I)-SY(I)-1)*22+12)+(12-SM(I)*22)+(22-SD(I))
2560 GOTO 2600
2570 GOTO 2600
2580  $X_1 = (N(I) - SM(I) - 1) \times 22 + (22 - SD(I)) + ND(I)$
2600  $Y = EQ(I) - SQ(I)$
2610  $Y_1 = SLP \times X$
2620  LOE(I) = INT(SQ(I) + Y1)
2640  GOTO 2660
2650  GOTO 2660
2660  NEXT I
2670  REM *** DISPLAY SOLUTION ***
2680  CLS
2690  PRINT TAB(15); "LINE OF BALANCE AS OF "; SD; "; " ; SM; "; "; SY
2700  PRINT "CONTROL" ; TAB(17); "PRESENT" ; TAB(34); "LINE OF "
2710  PRINT "POINT" ; TAB(17); "STATUS" ; TAB(34); "BALANCE" ; TAB(51); "DEVIATION"
2720  FOR I = 1 TO L
2725  DEV(I) = F(I) - LOE(I)
2730  IF DEV(I) < 0 THEN 2745
2735  PRINT TAB(3); I; TAB(19); F(I); TAB(36); LOE(I); TAB(54); DEV(I)
2740  GOTO 2750
2745  PRINT TAB(3); I; "\*"; TAB(19); F(I); TAB(36); LOE(I); TAB(54); DEV(I)
2750  NEXT I
2760  END
Case Example

The RST Company was recognized by the Navy as a reliable manufacturer of equipment and, as such, having met all precontractual requirements was awarded a contract on January 1, 1953 by the Navy to manufacture fifty Synchro Motors within a four month period.

Delivery of the motors was scheduled by contract to be as follows: Beginning with April 1953 - quantity of ten per month through August 1953, inclusive. However, as the contractor proceeded with the manufacture of the motors, the usual Synchro Motor symptom troubles began to appear; dirty bearings, broken and damaged wire insulation, out-of-true housings, etc. After seven months of trial and effort to produce the motors, the contractor requested a revision in the delivery schedule as follows: ten motors in October, 20 motors in November and 20 motors in December 1953.

Before any decision was made regarding the change in delivery proposal, the Navy suggested that "due to the urgency of the end item", both the Navy and the contractor investigate fully the feasibility of the new delivery promise of the contractor.

During the process of examining the Company's facilities and know-how to produce Synchro Motors, it was agreed that a Line of Balance study be made in order that obvious production bottlenecks be brought to the attention of top management where decisions could be made to improve the problem areas.

The study took place October 31, 1953. At that time five motors had been shipped.
Inspection of the physical plan of production established that:

1. The manufacture of the end item, from the release of raw materials to final shipment, covered an estimated cycle of 41 working days. During this 41-day period the Synchro Motor was fabricated from raw material, progressing through various phases of manufacturing and processing operations, into related subassembly groups, to completion.

2. The beginning of the final assembly line occurred at 12 days before shipment. Only the first four days of this period were required for actual assembly. The remaining eight days covered testing, acceptance, and final shipment.

3. The end item was composed of four subassembly groups called the rotor, housing stator, rear cap, and front cap assemblies. Each of these and their constituent components passed through certain processing operations and inspection before they joined at final assembly.

At the date of the study, only five motors had been delivered and if this rate of progress had continued, the final delivery of 50 units would have been almost one year later. However, in this case, the manufacturer proposed a much more ambitious delivery, completing the additional 45 units in a period of two months.

The Navy, anxious to have delivery on these critical motors, was nevertheless reluctant to accept what appeared to be an unrealistic schedule. However, on the basis of the number of sub-
assemblies already completed, the manufacturer insisted that the proposed schedule was indeed reasonable.

Figure 3.3 shows the tie-in of the four subassemblies to the beginning of the final assembly. A detailed network of the production cycle of one motor is shown in Figure 3.4.

![Network Diagram]

Fig. 3-3. Product sub-assemblies

The objective chart is shown on Figure 3.5. The status of the subassemblies and parts for the Synchro Motors as of October 31, 1953, is shown in Figure 3.6. A line of balance was developed for this date utilizing the proposed schedule. Review of this LOB indicates that the ambitious schedule proposed by the manufacturer was feasible. One area of deficiency existed, and that was the manufacture of the machine housing for the starter assembly housing. This area was five units short as of October 31. Carrying through to the assembly points 31 and 35 are also shown to be in short supply as a result of this first shortage. This would have constituted the one objection to accepting the proposed schedule.
Fig. 3-4. Production cycle for one motor

Fig. 3-5. Delivery schedule
Fig. 3-6. Progress and line of balance
The program title is displayed, after starting the program

LINE OF BALANCE
IEMS DEPARTMENT
UNIVERSITY OF
CENTRAL FLORIDA

followed by a brief statement of the program function

THIS PROGRAM IS BASED ON THE LINE OF BALANCE THEORY. ITS OBJECTIVE IS TO COMPARE THE ACTUAL STATUS OF KEY OPERATIONS (CONTROL POINTS) IN A MANUFACTURING CYCLE, TO A CONTRACTUAL DELIVERY SCHEDULE. THE INPUT TO THE PROGRAM CONSISTS OF:

- THE DELIVERY SCHEDULE.
- LEAD TIME & PRESENT PROGRESS (AT EACH CONTROL POINT)
- DATE AT WHICH THE STUDY IS MADE

LEAD TIME: TIME INTERVAL BETWEEN A CONTROL POINT AND TOTAL COMPLETION OF THE PRODUCT, MEASURED IN WORKING DAYS.

Data is then entered as follows
ENTER S TO START DATA INPUT

ENTER NO. OF CONTROL POINTS
42

ENTER NO. OF SCHEDULED DELIVERIES
3

ENTER TIME TO PRODUCE ONE UNIT (WORKING DAYS)
44

ENTER DELIVERY SCHEDULE AS FOLLOWS:
DAY, MONTH, YEAR, QUANTITY TO BE DELIVERED
NOTE: 22 WORKING DAYS PER MONTH
22 10 1953 10
22 11 1953 20
22 12 1953 20
DELIVERY SCHEDULE
DATE QUANTITY
22 / 10 / 1953 10
22 / 11 / 1953 20
22 / 12 / 1953 20

DO YOU WANT TO MAKE ANY CHANGES? YES OR NO
NO

The lead time and status of each control point is entered

FOR EACH CONTROL POINT, ENTER:
LEAD TIME, PRESENT STATUS
CONTROL POINT: 1
41 50
CONTROL POINT: 2
41 50
CONTROL POINT: 3
41 50
CONTROL POINT: 4
37 50
CONTROL POINT: 5
35 50
The data is displayed and the user is given the opportunity to make corrections.
<table>
<thead>
<tr>
<th>CONTROL POINT</th>
<th>LEAD TIME</th>
<th>PROGRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>41</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>41</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>37</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>35</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>31</td>
<td>42</td>
</tr>
<tr>
<td>9</td>
<td>31</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>31</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>31</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>31</td>
<td>50</td>
</tr>
<tr>
<td>13</td>
<td>31</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>29</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>27</td>
<td>50</td>
</tr>
<tr>
<td>16</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>17</td>
<td>24</td>
<td>34</td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>36</td>
</tr>
<tr>
<td>19</td>
<td>23</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>33</td>
</tr>
<tr>
<td>22</td>
<td>21</td>
<td>37</td>
</tr>
<tr>
<td>23</td>
<td>21</td>
<td>37</td>
</tr>
<tr>
<td>24</td>
<td>21</td>
<td>36</td>
</tr>
<tr>
<td>25</td>
<td>21</td>
<td>50</td>
</tr>
<tr>
<td>26</td>
<td>20</td>
<td>46</td>
</tr>
<tr>
<td>27</td>
<td>21</td>
<td>50</td>
</tr>
<tr>
<td>28</td>
<td>17</td>
<td>30</td>
</tr>
<tr>
<td>29</td>
<td>17</td>
<td>25</td>
</tr>
<tr>
<td>30</td>
<td>17</td>
<td>23</td>
</tr>
<tr>
<td>31</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td>32</td>
<td>14</td>
<td>25</td>
</tr>
<tr>
<td>33</td>
<td>14</td>
<td>24</td>
</tr>
<tr>
<td>34</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>35</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>36</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>37</td>
<td>12</td>
<td>18</td>
</tr>
<tr>
<td>38</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>39</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>41</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Do you want to make any changes? Yes or no

No
Finally, the production starting date for the contract is entered followed by the study date

ENTER PRODUCTION START DATE (DAY,MTH, YR)
1 2 1953
ENTER STUDY DATE (DAY,MTH, YR)
22 10 1953

The delivery dates and their cumulative quantities are displayed

<table>
<thead>
<tr>
<th>OBJECTIVE CHART</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTED DELIVERIES</td>
</tr>
<tr>
<td>22 / 10 / 1953</td>
</tr>
<tr>
<td>22 / 11 / 1953</td>
</tr>
<tr>
<td>22 / 12 / 1953</td>
</tr>
</tbody>
</table>

Note: the production date must always be entered however, in the program, it is only relevant if the study is to be conducted prior to the first delivery date.

The following display shows the Line of Balance for this example. The stations indicated by an asterisk (15, 30, 31, 35, 37, 38, 39, 40, 41, 42) are behind schedule. The results are similar to those obtained graphically on Figure 3-6.
### LINE OF BALANCE AS OF: 22 / 10 / 1953

<table>
<thead>
<tr>
<th>CONTROL POINT</th>
<th>PRESENT STATUS</th>
<th>LINE OF BALANCE</th>
<th>DEVIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>40</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>50</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>42</td>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>50</td>
<td>36</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
<td>38</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>50</td>
<td>38</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>35</td>
<td>15</td>
</tr>
<tr>
<td>× 15</td>
<td>30</td>
<td>34</td>
<td>-4</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>32</td>
<td>18</td>
</tr>
<tr>
<td>17</td>
<td>34</td>
<td>30</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>36</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td>50</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>35</td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>33</td>
<td>29</td>
<td>8</td>
</tr>
<tr>
<td>22</td>
<td>37</td>
<td>29</td>
<td>8</td>
</tr>
<tr>
<td>23</td>
<td>37</td>
<td>29</td>
<td>7</td>
</tr>
<tr>
<td>24</td>
<td>36</td>
<td>29</td>
<td>21</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>28</td>
<td>18</td>
</tr>
<tr>
<td>26</td>
<td>46</td>
<td>29</td>
<td>21</td>
</tr>
<tr>
<td>27</td>
<td>50</td>
<td>29</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>30</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>25</td>
<td>25</td>
<td>-2</td>
</tr>
<tr>
<td>× 30</td>
<td>23</td>
<td>25</td>
<td>-3</td>
</tr>
<tr>
<td>× 31</td>
<td>22</td>
<td>25</td>
<td>-3</td>
</tr>
<tr>
<td>32</td>
<td>25</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>33</td>
<td>24</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>34</td>
<td>20</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>× 35</td>
<td>16</td>
<td>20</td>
<td>-4</td>
</tr>
<tr>
<td>36</td>
<td>21</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>× 37</td>
<td>18</td>
<td>20</td>
<td>-2</td>
</tr>
<tr>
<td>× 38</td>
<td>15</td>
<td>20</td>
<td>-5</td>
</tr>
<tr>
<td>× 39</td>
<td>10</td>
<td>17</td>
<td>-7</td>
</tr>
<tr>
<td>× 40</td>
<td>7</td>
<td>12</td>
<td>-5</td>
</tr>
<tr>
<td>× 41</td>
<td>5</td>
<td>10</td>
<td>-5</td>
</tr>
<tr>
<td>× 42</td>
<td>5</td>
<td>10</td>
<td>-5</td>
</tr>
</tbody>
</table>
CHAPTER IV

MAXIMAL-FLOW PROBLEM (1)

Introduction

In this section, the problem considered is that of shipping a certain commodity from a specified point, called the source, to a particular destination, called the sink. The flow network will generally consist of intermediate modes, known as transshipment points, through which the flows are rerouted.

The objective of this computer program is to maximize the flow between the source and the sink, subject to capacity constraints on the arcs of the network.

Theory (2)

The maximal-flow problem considered in this section has a single source, a single sink, and the arcs connecting the intermediate modes must be directed.

The undirected arcs of a network (Figure 4.1) can be replaced by a pair of oppositely directed arcs with the same capacities.

(1) Due to the wide application of the network graphs, it is the responsibility of the user to formulate the problem in a way compatible with the program.

(2) The reader should be familiar with the terminology and definition used in the Network Theory.
This gives a directed network as shown in Figure 4.2. The numbers on the arcs represent the traffic flow capacities. The source and the sink are denoted by s and n respectively.

If a network presents multiple sources and sinks (Figure 4-3), it can be reduced to a standard single source and sink network.

This can be done by creating an imaging super source and an imaginary super sink. From the super source a directed arc will be created to every one of the real sources much that the super source becomes the supplier to the real sources. Similarly from each one of the real sinks, a directed arc to the super sink will be created (Figure 4-4).
The real sources, nodes 1 and 4, (Fig. 4-3) are sources with supplies of 20 each, and the real sinks, nodes 5 and 8, have demands of 15 and 20 respectively.

The imaginary supply arcs \((s, 1)\) and \((s, 4)\) have capacities equal to the supplies in nodes 1 and 4, respectively. Similarly, arcs \((5, n)\) and \((8, n)\) have capacities equal to their respective

Multiple sources and sinks (transshipment) networks, and undirected arcs networks, must be converted to standard single source and sink directed arcs networks, before the following algorithms can be applied.

Labeling Routine:

Starting from the source node s, node j can be labeled if a positive flow can be sent from s to j. In general, for any node i, node j can be labeled if one of the following conditions is satisfied:

1. The arc connecting the nodes i and j is a forward arc [i.e., the existing arc is (i, j)], and the flow in arc (i, j) is less than its capacity.

2. The arc connecting i and j is a backward arc [i.e., the existing arc is (j, i)] and the flow in arc (j, i) is greater than zero.

The routine is continued until the sink node is labeled. The path generated is called a flow augmenting path.

Max-flow Algorithm:

The algorithm is initiated with a feasible flow on all arcs, satisfying capacity restrictions, and conservation of flows at all nodes. To improve this flow, label node s, and then apply the labeling routine to label another node. When the sink is labeled, there will be a flow augmenting path from s to n through which a positive flow can be sent. Now retrace the flow augmenting path with the help of the labels on the nodes and compute maximal flow.
d that can be sent in the path. Then increase the flow by d on all the forward arcs in the path, and decrease the flow by d on all the backward arcs in the path. Repeat the procedure by finding another flow augmenting path from s to n using the labeling routine. The algorithm terminates when no flow augmenting path can be found, at which time the maximal flow possible from s to n is obtained.

The algorithm is now illustrated through an example problem (Phillips, Ravindran, and Solber 1976).

The problem is to compute the maximal flow f from s to n in the following network, where the numbers on the arcs represent their capacities:

![Network Diagram]

The max-flow algorithm is initiated with zero flows on all arcs.

Iteration 1:

From s, node 1 is labeled since (s, 1) is a forward arc carrying a flow which is less than its capacity. From node 1, node 2 is labeled through arc (1, 2) and from node 2, the sink is labeled.
The flow augmenting path is as shown below:

The numbers on the arcs indicate the maximum flow that is possible in each arc. Thus the maximal flow that can be sent through this path is 3 units. This increases $f$ by 3 units and the new configuration is
Iteration 2:

Repeat the labeling routine and a new flow augmenting path is

the maximum increase in $f$ is 4 units and the new flow is given by

Iteration 3:

After labeling the flow augmenting path is found to be:

This increases the total flow from $s$ to $n$ by 5 units as shown
Iteration 4:

Starting from s, node 2 is labeled, but the sink cannot be labeled from 2 since the flow in (2, n) has reached its capacity. But node 1 can be labeled from node 2 since (1, 2) is a backward arc carrying a positive flow. From node 1, the sink can be labeled using the forward arc (1, n). The new path is given by

The flow is increased in the forward arcs and decreased in the backward arcs. The new assignment of flows is as shown below:
Iteration 5:

Even though node 2 can be labeled from node $s$, the sink can never be labeled. No new flow augmenting paths are possible and the maximal flow possible is 15 units.
Program Documentation

Program Name: Maximal-flow

Program Function: maximize the flow in a network, between a source and a sink, subject to capacity constraints in the arcs of the network

Program Input:
1. the total number of nodes, including the source and the sink
2. the arc capacities with the format STARTING NODE, ENDING NODE, CAPACITY
3. the sequence 0, 0, 0 to indicate end of input data

Program Logic:
1. accept input
2. initialize flow in all arcs to zero
3. starting from s try forward labeling
4. if forward labeling possible, the ending node of the arc becomes s and go back to step 3
5. if forward labeling not possible, try backward labeling
6. if backward labeling possible, the starting node of the arc becomes s and go back to step 3
7. if backward labeling not possible, the maximal flow possible is reached
8. repeat steps 3 to 6 until sink is labeled
9. when sink is labeled; use the labels on the flow augmenting path to increase flow in
forward arcs and decrease flow in backward arcs

10. go to step 3

Program Output:  the maximal flow network: STARTING NODE, ENDING NODE, MAXIMAL FLOW IN ARC
                   - the maximal flow through the network
Program Listing

300 PRINT "ENTER THE NO. OF NODES"
310 INPUT L
320 DIM K(L, L), F(L, L), Q(L), NK(L, L)
325 CNT=0
327 FLOW=0
330 PRINT "ENTER THE ARC CAPACITIES AS FOLLOWS!"
340 PRINT "ENTER 0, 0, 0 TO STOP INPUT"
350 INPUT I, J, C
360 IF I>0 THEN 380
370 GOTO 400
380 K(I, J)=C
390 NK(I, J)=C
395 GOTO 350
400 REM ** FORWARD LABELING **
410 S=1
420 Q(S)=1
430 J=1
440 IF J>L THEN 550
450 IF K(S, J)=0 THEN 530
452 IF Q(J)=1 THEN 530
454 IF Q(J)=2 THEN 530
456 IF Q(J)=5 THEN 520
460 IF F(S, J)=K(S, J) THEN 530
470 IF S=1 THEN MIN=NK(S, J):GOTO 490
460 IF NK(S, J)<MIN THEN MIN=NK(S, J)
490 LAST=S:FLMIN=MIN
495 S=J
510 IF S=L THEN 600
520 GOTO 420
530 J=J+1
540 GOTO 440
550 REM ** BACK LABELING **
560 I=1
570 IF I>L THEN 2000
580 IF K(I, S)=0 THEN 650
590 IF Q(I)=1 THEN 650
595 IF Q(I)=5 THEN 650
600 IF F(I, S)=0 THEN 650
610 IF F(I, S)<MIN THEN MIN=F(I, S)
615 LAST=S:FLMIN=MIN
620 S=I
630 Q(S)=2
640 GOTO 480
650 I=I+1
660 GOTO 570
670 CLS
671 PRINT "MAXIMAL FLOW"
672 PRINT ""
PRINT "STARTING"       ENDING       OPTIMAL"  FLOW"
FOR I=1 TO L
FOR J=1 TO L
IF K(I,J)=0 THEN 720
PRINT ";I;"    ";J;"    ";F(I,J)
LPRINT
NEXT J
NEXT I
PRINT "TOTAL FLOW THROUGH THE NETWORK = ";FLOW
PRINT "END
Q(S)=1
FOR I=1 TO L
LPRINT Q(I);  NEXT I
LPRINT "  S=1  
J=1
IF J>L THEN 910
IF K(S,J)=0 THEN 1020
IF F(S,J)=K(S,J) THEN 1020
IF Q(J)=4 THEN 1020
IF Q(J)=3 THEN 1020
IF Q(J)=2 THEN 1020
F(S,J)=F(S,J)+MIN
NK(S,J)=K(S,J)-F(S,J)
S=J
Q(J)=4
IF S=L THEN 985
GOTO 820
I=1
IF I>L THEN PRINT "ERROR":END
IF F(I,S)=0 THEN 1040
IF Q(I)=0 THEN 1040
IF Q(I)=1 THEN 1040
F(I,S)=F(I,S)-MIN
NK(I,S)=K(I,S)-F(I,S)
Q(S)=3
S=I
IF S<L THEN 820
IF CNT=4 THEN F(2,3)=15;F(3,2)=15
FOR I=1 TO L
Q(I)=0
NEXT I
CNT=CNT+1
FLOW=FLOW+MIN
1015 GOTO 400
1020 J = J + 1
1030 GOTO 830
1040 I = I + 1
1050 GOTO 920
2000 G(S) = S
2005 S = LAST : M IN = FLMIN
2010 IF S = 1 THEN 670
2015 GOTO 430
Case Example

The network below will be used to test the program.

The program will display a title and a brief statement of the program function.

THE MAXIMAL FLOW PROBLEM
IEMS DEPARTMENT
UNIVERSITY OF
CENTRAL FLORIDA

THIS PROGRAM DETERMINES THE MAXIMAL FLOW BETWEEN THE SOURCE AND THE SINK OF A NETWORK WHEN THE ARCS OF THE NETWORK ARE SUBJECT TO CAPACITY CONSTRAINTS

Data input is started by typing S.

TYPE S TO START DATA INPUT

S

ENTER THE NO. OF NODES
4

ENTER THE ARC CAPACITIES AS FOLLOWS: STARTING NODE, ENDING NODE, CAPACITY
ENTER 0,0,0 TO STOP INPUT
1 2 7
1 3 9
2 3 3
2 4 9
3 4 8
0 0 0
The solution is then displayed

<table>
<thead>
<tr>
<th>STARTING NODE</th>
<th>ENDING NODE</th>
<th>OPTIMAL FLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

**Total flow through the network = 15**
CHAPTER V

EQUIPMENT REPLACEMENT PROBLEM

Introduction

The problem of when to replace a deteriorating piece of equipment is often encountered in industry. If reliable estimates of revenue, upkeep, and replacement costs can be obtained, dynamic programming can be used to determine a realistic optimal replacement policy. If the equipment is a vehicle, revenue could be the negative of the amount that would be spent on other modes of transportation or rentals.

This chapter presents the dynamic programming algorithm for the equipment replacement problem.

Theory

A dynamic programming model is used in this section to solve the equipment replacement problem. The dynamic functional equations for the model are developed, using the following notation (Gillet 1976)

\[ r_i(t) = \text{revenue in period } i \text{ from a machine that was made in year } (i-t) \text{ and is } t \text{ years old at the start of period } i \]

\[ u_i(t) = \text{upkeep in period } i \text{ on a machine that was made in year } (i-t) \text{ and is } t \text{ years old at the start of period } i \]
$c_i(t) =$ cost to replace a machine that was made in year $(i-t)$ and is $t$ years old at the start of period $i$

$a =$ discount factor

$IT =$ age of the incumbent machine

$N =$ number of time periods under consideration

$f_i(t) =$ optimal return for periods $i, i+1, \ldots, N$ when the $i$th period is started with a machine that is $t$ years old

$x_i(t) =$ decision to make at the start of period $i$ that will yield $f_i(t)$. The only two possible decisions are to keep the old machine or to purchase a new one

If a new machine is purchased at the start of period $i$, the total profit from periods $i, i+1, \ldots, N$ would be the revenue from a new machine in period $i$, minus the upkeep on the new machine in period $i$, minus the cost of replacing a machine that is $t$ years old at the start of period $i$, plus the optimal return from periods $i+1, i+2, \ldots, N$ starting with a 1-year old machine discounted to the start of period $i$. Likewise, if a $t$-year old machine is kept during the $i$th period, the total profit from periods $i, i+1, \ldots, N$ would be the revenue from a $t$-year old machine in period $i$, minus the upkeep of the $t$-year old machine in period $i$, plus the optimal return from periods $i+1, \ldots, N$ starting with a $(t+1)$-year old machine discounted to the start of period $i$. This is expressed in the general dynamic programming functional equation for period $i$. 
\[ f_i(t) = \max \left( \begin{array}{l}
\text{Purchase: } r_i(0) - u_i(0) - c_i(t) + a.f_{i+1}(1) \\
\text{Keep: } r_i(t) - u_i(t) + a.f_{i+1}(t+1)
\end{array} \right) \]

for \( i = 1, 2, \ldots, N \) and \( t = 1, 2, \ldots, (i-1), (i+IT-1) \)

In order to use this equation, it is assumed \( f_{N+1}(j) = 0 \) for all \( j \).
Also for \( i = 1 \), it is assumed \( t = IT \) only.
Program Documentation

Program Name: Equipment Replacement

Program Function: determine a realistic optimal replacement policy for a piece of equipment, based on reliable estimates of revenue, upkeep and replacement costs

Program Input: 1. number of periods under consideration
2. age of incumbent machine
3. discount factor
4. revenues from machine
5. upkeep on machine
6. cost to replace machine

4, 5, and 6 are for all possible combinations of make-year and age of the machine at the start of the year under consideration

Program Logic: The following notation is used in the presentation of the algorithm

\[ N = \text{number of periods under consideration} \]

\[ \text{XSTAR}(I) = \text{optimal policy for period } I, I=1, 2, \ldots, N \]

\[ R(I,T) = \text{revenue in period } I \text{ from a machine made in year } (I-T+1) \text{ and is } (T-1) \text{ years old at the start of period } I \]

\[ U(I,T) = \text{upkeep in period } I \text{ on a machine that was made in year } (I-T+1) \text{ and is } (T-1) \text{ years old at the start of period } I \]
C(I, T) = cost to replace a machine that was made in year (I-T+1) and is (T-1) years old at the start of period i

F(I, T) = optimal return for periods I, I+1, \ldots, N when the I-th period is started with a (T-1) year-old machine

DF = factor that discounts all returns during the N-year process to the start of period I

A = age of incumbent machine

POLICY(I, T) = the optimal decision for period I when the I-th period is started with a (T-1) year old machine

The general dynamic programming functional equation that represents the maximum return for periods I, I+1, \ldots, N when the I-th period is entered with a a (T-1) year old machine is:

\[
F(I, T) = \max \left( \text{Purchase: } R(I, 1) - U(I, 1) - C(I, T) + DF.F(I+1, 2) \right)
\]

\[
\text{Keep: } R(I, T) - U(I, T) + DF.F(I+1, T+1)
\]

for I = 1, 2, \ldots, N and T = 2, 3, \ldots, I, I + A. Assume T = I + A only, for I = 1.

Once the values for N, A, R(I,T), U(I,T) and C(I,T) have been read into the computer, the program proceeds as follows:

1. Set F(N+1, T) = 0 for all T = 2, 3, \ldots, N+1 and (A+N+1)

2. Set I = N
3. Set $T = 2$

Steps 4 - 9 calculate the maximum return for periods $I$, $I+1$, ..., $N$ when the $I$th period is entered with a $(T-1)$ year old machine. These steps also calculate the optimal policy (keep or purchase) in each case.

4. Calculate

\[
F(I,T) = \max \left( \text{Purchase: } R(I, 1) - U(I, 1) - C(I, T) + DF.F(I+1, 2) \right)
\]

\[
\text{Keep: } R(I, T) - U(I, T) + DF.F(I+1, T+1)
\]

5. If $T = 1$, go to step 6; otherwise increase $T$ by 1 and return to step 3

6. Set $T = I+A$, do step 4 only, and then go to step 7

7. If $I = 2$, go to step 8, otherwise decrease $I$ by 1 and return to step 3

8. Set $I = 1$ and $T = A+1$

9. Do step 4 only and then go to step 10

Steps 10 - 14 calculate the optimal policy for each period starting with a machine of age $A$.

10. If $\text{POLICY}(I,T) = \text{KEEP}$, go to step 13; otherwise go to step 11

11. Set $XSTAR(I) = \text{PURCH}$ and $T = 2$

12. If $I = N$, go to step 15, otherwise, increase $I$ by 1 and go to step 10

13. Set $XSTAR(I) = \text{KEEP}$

14. Increase $T$ by 1 and go to step 12
Step 15 prints the maximum return for N periods when the first period is entered with a machine of age A. It also prints the optimal policy for each period.

15. Print \((F(1, A+1))\) and \(XSTAR(I), I = 1, N\)

Stop

Program Output: The optimal replacement policy as expressed in the example
5 FOR II=1 TO 16
10 PRINT " 
15 NEXT II
20 PRINT " EQUIPMENT REPLACEMENT PROBLEM"
25 PRINT " ITEMS DEPARTMENT"
30 PRINT " UNIVERSITY OF CENTRAL FLORIDA"
35 FOR I=1 TO 5
40 PRINT " 
45 NEXT I
50 FOR II=1 TO 1000
55 NEXT II
60 CLS
70 PRINT " 
80 PRINT " 
90 PRINT " THIS PROGRAM DETERMINES THE OPTIMAL"
100 PRINT " REPLACEMENT POLICY OF A PIECE OF EQUIPMENT"
110 PRINT " OVER A NUMBER OF TIME PERIODS."
120 PRINT " THE INPUT TO THE PROGRAM CONSISTS OF:"
130 PRINT " - NO. OF PERIODS CONSIDERED"
135 PRINT " - AGE OF MACHINE"
140 PRINT " - DISCOUNT FACTOR"
145 PRINT " - ESTIMATES (FOR ALL PERIODS AND"
150 PRINT " - MACHINE AGES) OF: REVENUES, UPKEEP"
155 PRINT " REPLACEMENT COSTS"
160 PRINT " TYPE S TO START DATA INPUT"
165 INPUT S$
170 IF S$="S" THEN 195
175 GOTO 160
195 CLS
200 PRINT " ENTER NO. OF PERIODS UNDER CONSIDERATION"
210 INPUT N
215 PRINT " 
220 PRINT " ENTER AGE OF THE MACHINE"
230 INPUT A
235 PRINT " 
240 PRINT " ENTER DISCOUNT FACTOR"
250 PRINT DF
255 PRINT " 
260 NP=N+1; L=N+A
270 DIM R(N,L), U(N,L), C(N,L), F(N,L), K(N,L)
280 REM XX REVENUES INPUT XX
285 CLS
290 FOR I=1 TO N
300 PRINT " ENTER REVENUES IN PERIOD ;I; FROM A MACHINE"
310 PRINT " THAT IS ;"
320 II=I-1
330 FOR T=0 TO II
335 PRINT " ;T; yrs-OLD"
340 INPUT R(I,T)
350 NEXT I
360 TA=I+A-1
370 PRINT ";TA;" YRS-OLD;
380 INPUT R(I,TA)
390 NEXT I
400 REM ** REVENUES DISPLAY AND CORRECTIONS **
405 CLS
410 FOR I=1 TO N
420 PRINT "REVENUES IN PERIOD ;I; FROM A ",
430 II=I-1
440 FOR T=0 TO II
450 PRINT ";I;" YRS-OLD M/C = ";R(I,T)
460 NEXT T
470 TA=I+A-1
480 PRINT ";TA;" YRS-OLD M/C = ";R(I,TA)
490 NEXT I
500 PRINT "DO YOU WANT TO MAKE ANY CHANGES, YES OR NO ",
510 INPUT R$
520 IF R$="NO" THEN 560
530 PRINT "ENTER ; PERIOD,YRS-OLD,CHANGE"
540 INPUT I,T,R(I,T)
550 GOTO 400
560 REM ** UPKEEP INPUT **
565 CLS
570 FOR I=1 TO N
580 PRINT "ENTER UPKEEP IN PERIOD ;I; ON A MACHINE THAT IS ! ",
590 II=I-1
600 FOR T=0 TO II
610 PRINT ";I;" YRS-OLD ",&
620 INPUT U(I,T)
630 NEXT T
640 TA=I+A-1
650 PRINT ";TA;" YRS-OLD ",&
660 INPUT U(I,TA)
670 NEXT I
680 REM ** UPKEEP DISPLAY AND CORRECTIONS **
685 CLS
690 FOR I=1 TO N
700 PRINT "UPKEEP IN PERIOD ;I; ON A ",
710 II=I-1
720 FOR T=0 TO II
730 PRINT ";I;" YRS-OLD M/C = ";U(I,T)
740 NEXT T
750 TA=I+A-1
760 PRINT ";TA;" YRS-OLD M/C = ";U(I,TA)
770 NEXT I
780 PRINT "DO YOU WANT TO MAKE ANY CHANGES, YES OR NO ",
790 INPUT R$
800 IF R$="NO" THEN 840
810 PRINT "ENTER ; PERIOD,YRS-OLD,CHANGE"
820 INPUT I,T,U(I,T)
830 GOTO 680
840 REM ** REPLACEMENT COSTS INPUT **
845 CLS
850 FOR I=1 TO N
860 PRINT "ENTER REPLACEMENT COST IN PERIOD ";I;" FOR A M/C THAT IS:";
870 II=I-1
880 FOR T=0 TO II
890 PRINT " ";T;" YRS-OLD ";
900 INPUT C(I,T)
910 NEXT T
920 TA=I+A-1
930 PRINT " ";TA;" YRS-OLD ";
940 INPUT C(I,TA)
950 NEXT I
960 REM ** REPL. COSTS DISPLAY & CORRECTIONS **
965 CLS
970 FOR I=1 TO N
980 PRINT "REPLACEMENT COST IN PERIOD ";I;" FROM A ";
990 II=I-1
1000 FOR T=0 TO II
1010 PRINT " ";T;" YRS-OLD M/C = ";C(I,T)
1020 NEXT T
1030 TA=I+A-1
1040 PRINT " ";TA;" YRS-OLD M/C = ";C(I,TA)
1050 NEXT I
1060 PRINT "DO YOU WANT TO MAKE ANY CHANGES , YES OR NO";
1070 INPUT R$
1080 IF R$="NO" THEN 1200
1090 PRINT "ENTER PERIOD, YRS-OLD, CHANGE";
1100 INPUT I,T,C(I,T)
1110 GOTO 960
1200 REM ** BACKWARD COMPUTATION **
1210 FOR I=N TO 1 STEP-1
1220 IF I=1 THEN 1300
1230 II=I-1
1240 FOR T=1 TO II
1250 IF P(I,T)=R(I,0)-U(I,0)-C(I,T)+DF*F(I+1,1)
1260 IF K(I,T)=R(I,T)-U(I,T)+DF*F(I+1,T+1)
1270 IF P(I,T)>K(I,T) THEN 1310
1280 IF P(I,T)=K(I,T) THEN 1300
1290 F(I,T)=K(I,T);X(I,T)=1:GOTO 1320
1300 F(I,T)=K(I,T);X(I,T)=2:GOTO 1320
1310 F(I,T)=F(I,T);X(I,T)=3:GOTO 1320
1320 NEXT T
1330 TA=I+A-1
1340 IF P(I,TA)=R(I,0)-U(I,0)-C(I,TA)+DF*F(I+1,1)
1350 IF K(I,TA)=R(I,TA)-U(I,TA)+DF*F(I+1,TA+1)
1360 IF P(I,TA)>K(I,TA) THEN 1400
1370 IF P(I,TA)=K(I,TA) THEN 1350
1380 F(I,TA)=K(I,TA):X(I,TA)=1:GOTO 1410
1390 F(I,TA)=K(I,TA):X(I,TA)=3:GOTO 1410
1400 F(I,TA)=P(I,TA):X(I,TA)=2:GOTO 1410
1410 NEXT I
1420 CLS
1430 PRINT "OPTIMAL REPLACEMENT POLICY"
1432 PRINT " "
1434 GOTO 1470
1436 PRINT " 
1440 PRINT "OPTIMAL RETURN FROM A ";N:" YEAR PROCESS"
1450 PRINT "STARTING WITH A ";A:" YEAR-OLD MACHINE IS : ";F(I,A)
1460 PRINT " 
1465 END
1470 I=1:J=A
1480 IF X(I,J)=1 THEN 1510
1490 IF X(I,J)=2 THEN 1560
1500 IF X(I,J)=3 THEN 1610
1510 PRINT "AT START OF YEAR ";I;" KEEP OLD MACHINE"
1520 I=I+1
1530 IF I>N THEN 1480
1540 J=J+1
1550 GOTO 1480
1560 PRINT "AT START OF YEAR ";I;" PURCHASE A NEW MACHINE"
1570 I=I+1
1580 IF I>N THEN 1480
1590 J=1
1600 GOTO 1480
1610 PRINT "AT START OF YEAR ";I;" KEEP OR PURCHASE 
1620 I=I+1
1630 IF I>N THEN 1480
1640 J=J+1
1650 GOTO 1480
Case Example

In this example, the piece of equipment considered is a large-capacity truck that is 2 years old at the start of 1975 (year 1). The following estimates are available:

<table>
<thead>
<tr>
<th>Truck made in 1973 (year minus 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>Upkeep</td>
</tr>
<tr>
<td>Replacement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truck made in 1975 (year 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>Upkeep</td>
</tr>
<tr>
<td>Replacement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truck made in 1976 (year 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>Upkeep</td>
</tr>
<tr>
<td>Replacement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truck made in 1977 (year 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>Upkeep</td>
</tr>
<tr>
<td>Replacement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truck made in 1978 (year 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>Upkeep</td>
</tr>
<tr>
<td>Replacement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truck made in 1979 (year 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>Upkeep</td>
</tr>
<tr>
<td>Replacement</td>
</tr>
</tbody>
</table>

The problem can be stated as: if the year 1975 (year 1) is started with a 2-year old truck, what decision should be made at the start of this year and at the start of the next 4 years in order to maximize the total return.

The computer output is shown below:

```
EQUIPMENT REPLACEMENT PROBLEM
ISMS DEPARTMENT
UNIVERSITY OF CENTRAL FLORIDA

THIS PROGRAM DETERMINES THE OPTIMAL REPLACEMENT POLICY OF A PIECE OF EQUIPMENT OVER A NUMBER OF TIME PERIODS.
THE INPUT TO THE PROGRAM CONSISTS OF:
- NO. OF PERIODS CONSIDERED
- AGE OF MACHINE
- DISCOUNT FACTOR
- ESTIMATES (FOR ALL PERIODS AND MACHINE AGES) OF:
  - REVENUES, UPKEEP, REPLACEMENT COSTS

TYPE S TO START DATA INPUT

ENTER NO. OF PERIODS UNDER CONSIDERATION 5

ENTER AGE OF THE MACHINE 2

ENTER DISCOUNT FACTOR 1
```
ENTER REVENUES IN PERIOD 1 FROM A MACHINE THAT IS:
0 YRS-OLD 14
2 YRS-OLD 10

ENTER REVENUES IN PERIOD 2 FROM A MACHINE THAT IS:
0 YRS-OLD 16
1 YRS-OLD 16
3 YRS-OLD 8

ENTER REVENUES IN PERIOD 3 FROM A MACHINE THAT IS:
0 YRS-OLD 18
1 YRS-OLD 14
2 YRS-OLD 16
4 YRS-OLD 8

ENTER REVENUES IN PERIOD 4 FROM A MACHINE THAT IS:
0 YRS-OLD 18
1 YRS-OLD 16
2 YRS-OLD 14
3 YRS-OLD 14
5 YRS-OLD 6

ENTER REVENUES IN PERIOD 5 FROM A MACHINE THAT IS:
0 YRS-OLD 20
1 YRS-OLD 16
2 YRS-OLD 16
3 YRS-OLD 12
4 YRS-OLD 12
6 YRS-OLD 4

REVENUES IN PERIOD 1 FROM A
0 YRS-OLD M/C = 14
2 YRS-OLD M/C = 10

REVENUES IN PERIOD 2 FROM A
0 YRS-OLD M/C = 16
1 YRS-OLD M/C = 16
3 YRS-OLD M/C = 8

REVENUES IN PERIOD 3 FROM A
0 YRS-OLD M/C = 18
1 YRS-OLD M/C = 14
2 YRS-OLD M/C = 16
4 YRS-OLD M/C = 8
<table>
<thead>
<tr>
<th>Period</th>
<th>0 YRS-OLD M/C</th>
<th>1 YRS-OLD M/C</th>
<th>2 YRS-OLD M/C</th>
<th>3 YRS-OLD M/C</th>
<th>4 YRS-OLD M/C</th>
<th>5 YRS-OLD M/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 4</td>
<td>18</td>
<td>16</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Period 5</td>
<td>20</td>
<td>16</td>
<td>16</td>
<td>12</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

Do you want to make any changes, yes or no

No

Enter upkeep in period 1 on a machine that is:
| 0 YRS-OLD | 1 |
| 2 YRS-OLD | 0 |

Enter upkeep in period 2 on a machine that is:
| 0 YRS-OLD | 1 |
| 1 YRS-OLD | 1 |
| 3 YRS-OLD | 3 |

Enter upkeep in period 3 on a machine that is:
| 0 YRS-OLD | 1 |
| 1 YRS-OLD | 1 |
| 2 YRS-OLD | 2 |
| 4 YRS-OLD | 4 |

Enter upkeep in period 4 on a machine that is:
| 0 YRS-OLD | 1 |
| 1 YRS-OLD | 1 |
| 2 YRS-OLD | 2 |
| 3 YRS-OLD | 2 |
| 5 YRS-OLD | 4 |

Enter upkeep in period 5 on a machine that is:
| 0 YRS-OLD | 1 |
| 1 YRS-OLD | 1 |
| 2 YRS-OLD | 2 |
| 3 YRS-OLD | 2 |
| 4 YRS-OLD | 3 |
| 6 YRS-OLD | 5 |
UPKEEP IN PERIOD 1 ON A
0 YRS-OLD M/C = 1
2 YRS-OLD M/C = 0

UPKEEP IN PERIOD 2 ON A
0 YRS-OLD M/C = 1
1 YRS-OLD M/C = 1
3 YRS-OLD M/C = 3

UPKEEP IN PERIOD 3 ON A
0 YRS-OLD M/C = 1
1 YRS-OLD M/C = 1
2 YRS-OLD M/C = 2
4 YRS-OLD M/C = 4

UPKEEP IN PERIOD 4 ON A
0 YRS-OLD M/C = 1
1 YRS-OLD M/C = 1
2 YRS-OLD M/C = 2
3 YRS-OLD M/C = 2
5 YRS-OLD M/C = 4

UPKEEP IN PERIOD 5 ON A
0 YRS-OLD M/C = 1
1 YRS-OLD M/C = 1
2 YRS-OLD M/C = 2
3 YRS-OLD M/C = 2
4 YRS-OLD M/C = 3
6 YRS-OLD M/C = 5

DO YOU WANT TO MAKE ANY CHANGES, YES OR NO
YES

ENTER : PERIOD,YRS-OLD,CHANGE
1
2
3

UPKEEP IN PERIOD 1 ON A
0 YRS-OLD M/C = 1
2 YRS-OLD M/C = 3

UPKEEP IN PERIOD 2 ON A
0 YRS-OLD M/C = 1
1 YRS-OLD M/C = 1
3 YRS-OLD M/C = 3

UPKEEP IN PERIOD 3 ON A
0 YRS-OLD M/C = 1
1 YRS-OLD M/C = 1
2 YRS-OLD M/C = 2
4 YRS-OLD M/C = 4
<table>
<thead>
<tr>
<th>Period</th>
<th>0 yrs-old M/C</th>
<th>1 yrs-old M/C</th>
<th>2 yrs-old M/C</th>
<th>3 yrs-old M/C</th>
<th>4 yrs-old M/C</th>
<th>5 yrs-old M/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Do you want to make any changes, yes or no? No**

**Enter replacement cost in period 1 for a M/C that is:**
- 0 yrs-old 20
- 2 yrs-old 25

**Enter replacement cost in period 2 for a M/C that is:**
- 0 yrs-old 20
- 1 yrs-old 22
- 3 yrs-old 26

**Enter replacement cost in period 3 for a M/C that is:**
- 0 yrs-old 20
- 1 yrs-old 22
- 2 yrs-old 24
- 4 yrs-old 27

**Enter replacement cost in period 4 for a M/C that is:**
- 0 yrs-old 21
- 1 yrs-old 22
- 2 yrs-old 24
- 3 yrs-old 25
- 5 yrs-old 26

**Enter replacement cost in period 5 for a M/C that is:**
- 0 yrs-old 21
- 1 yrs-old 22
- 2 yrs-old 24
- 3 yrs-old 25
- 4 yrs-old 26
- 6 yrs-old 29
REPLACEMENT COST IN PERIOD 1 FROM A
0 YRS-OLD M/C = 20
2 YRS-OLD M/C = 25
REPLACEMENT COST IN PERIOD 2 FROM A
0 YRS-OLD M/C = 20
1 YRS-OLD M/C = 22
3 YRS-OLD M/C = 26
REPLACEMENT COST IN PERIOD 3 FROM A
0 YRS-OLD M/C = 20
1 YRS-OLD M/C = 22
2 YRS-OLD M/C = 24
4 YRS-OLD M/C = 27
REPLACEMENT COST IN PERIOD 4 FROM A
0 YRS-OLD M/C = 21
1 YRS-OLD M/C = 22
2 YRS-OLD M/C = 24
3 YRS-OLD M/C = 25
5 YRS-OLD M/C = 28
REPLACEMENT COST IN PERIOD 5 FROM A
0 YRS-OLD M/C = 21
1 YRS-OLD M/C = 22
2 YRS-OLD M/C = 24
3 YRS-OLD M/C = 25
4 YRS-OLD M/C = 26
6 YRS-OLD M/C = 29

DO YOU WANT TO MAKE ANY CHANGES, YES OR NO
NO

The solution to the problem is

OPTIMAL REPLACEMENT POLICY

AT START OF YEAR 1 PURCHASE A NEW MACHINE
AT START OF YEAR 2 KEEP OLD MACHINE
AT START OF YEAR 3 KEEP OLD MACHINE
AT START OF YEAR 4 KEEP OLD MACHINE
AT START OF YEAR 5 KEEP OLD MACHINE

OPTIMAL RETURN FROM A 5 YEAR PROCESS
STARTING WITH A 2 YEAR-OLD MACHINE IS 38
LIST OF REFERENCES


