N-96-01
II-A-334

DOCUMENTATION
OF THE
EPA/ONAC STRATEGY MODEL

Strategy Model
Summary

Acronym: None
Media/subject: Noise

MODEL OVERVIEW: Determines the minimum cost mix of regulations to achieve level of reduction in noise or gives a cost limit to achieve the maximum reduction of noise.

FUNCTIONAL CAPABILITIES: The Strategy Model prioritizes the cost effectiveness of a given number of products being considered for noise regulations.

BASIC ASSUMPTIONS: Contact Dr. Kurt Askin for a description of the basic assumptions of this model.

INPUT: Costs of regulation of different types of machines at various noise levels and the benefits of regulation are the inputs to this model.

OUTPUT: Listing of different regulations to achieve a certain fixed level of noise reduction are the outputs of the model.

COMPUTATIONAL SYSTEM REQUIREMENTS:
Hardware: Mainframe IBM 370
Printer any model
Language: Fortran
Operator skills: Programming
economics

APPLICATIONS: The model has been used in noise regulation review to bring to management's attention cost effective options in a set of products regulations.

CONTACT: Kurt Askin U.S. EPA, Office of Noise Abatement and Control
Crystal Mall #2, ANR-471 PH: (703) 557-9300
1921 Jefferson Davis Hwy, Arlington, VA 22202

REFERENCES: Contact Dr. Kurt Askin for references describing the model.

TABLE OF CONTENTS

1. OVERVIEW OF THE MODEL

1. OVERVIEW OF THE MODEL

The Strategy Model has been developed for the Office of Noise Abatement and Control (ONAC) of the Environmental Protection Agency (EPA) by Science Applications, Inc. (SAI). The Model is designed to assist in the search for an optimal mix of regulatory options when a large number of product types are being regulated. The typical questions that the Model can help answer are the following:

- Given an upper bound on the total regulatory cost on all the product types, which is the set of regulatory options that will result in the maximum total benefits?
- Given a lower bound on the total benefits for all the product types, which is the set of regulatory options that will result in the minimum total regulatory cost?

The Strategy Model makes certain assumptions on the availability of the regulatory cost and benefit data and the functional relationship between these data for each product type under study. The cost and benefit corresponding to each regulatory option for each product type are assumed known. Thus, a set of discrete cost-benefit data points are available. The optimization procedure of the Model requires the cost-benefit relationship be given in a functional form. Therefore, the set of discrete cost-benefit points must be parametrized into a continuous cost-benefit functional relationship. This can be achieved using a least-square fit over the data with an assumed functional form. The Model further assumes that the costs and benefits of the product types are additive, and that the cost-benefit functions are convex, i.e. the marginal benefit to cost ratio decreases as cost increases. In practice, these assumptions are usually valid. Policy constraints in the form of upper and lower bounds on the cost and benefit of each product type can also be taken into account by the Model.

Using the cost-benefit functions, together with the policy constraints for each product type, an optimization procedure based on the

Lagrange multipliers method is carried out by the Strategy Model. This procedure gives the optimal mix of regulations for any level of total regulatory cost or benefit desired. The procedure has been computerized and the program is currently operational on EPA's computer system. The program outputs the optimal regulatory costs of each product type for different levels of total regulatory costs or benefits. From these regulatory costs, the optimal mixes of regulations or their approximations can be obtained. The Strategy Model is schematically represented in Figure 1.

In Section 2, the mathematical formulation of the Model is presented, together with the derivations of the key equations used. The input data required to run the Model and preparation of the data are discussed in Section 3. Descriptions and interpretations of the output data are given in Section 4. In Section 5, the last section, procedures for using the Strategy Model and steps for operating the computer program of the optimization routine are explained.

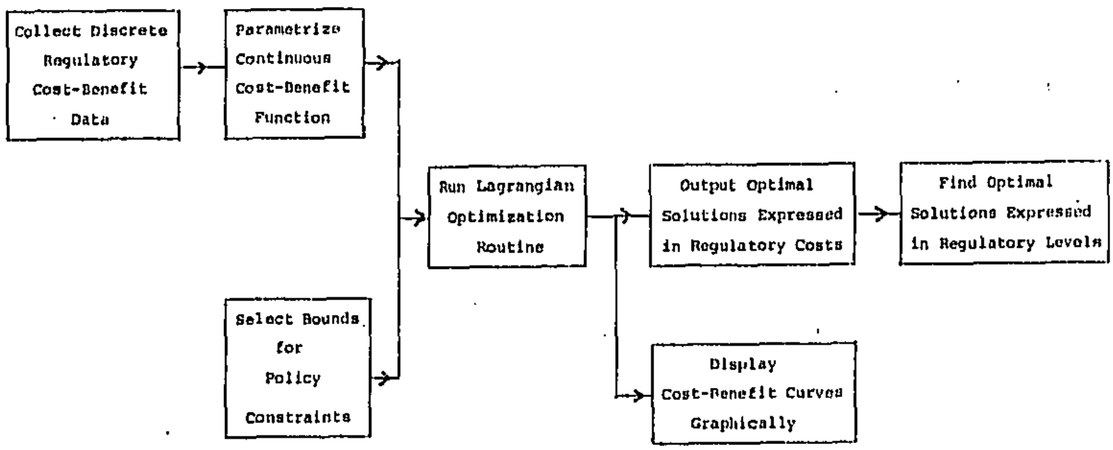


FIGURE 1. SCHEMATIC REPRESENTATION OF THE STRATEGY MODEL

2. MATHEMATICAL FORMULATION

2. MATHEMATICAL FORMULATION

There are two key components in the Strategy Model that contain mathematical treatments requiring detailed discussion. One is the parametrization of the cost-benefit curves. The other is the Lagrangian optimization routine. These are presented in the following subsections.

2.1 PARAMETRIZATION OF COST-BENEFIT CURVES

The Strategy Model requires the cost-benefit relationship for each product type to be given in a convex (or concave) functional form for a minimization (or maximization) problem. The typical cost-benefit curve shown in Figure 2 is found in many regulatory analyses and is assumed in the Model. The following equation is used to parametrize the cost-benefit function:

$$b = \frac{\alpha}{\beta + c} - \gamma$$

where b is the benefit,
c is the cost, and
α, β, γ are parameters.

At the origin, b = 0 and c = 0,

therefore, $b = \frac{\alpha}{\beta + c} - \frac{\alpha}{\beta}$. (1)

Rearranging, $$\left(\frac{1}{b}\right) = - \frac{\beta}{\alpha} - \frac{\beta^2}{\alpha} \left(\frac{1}{c}\right) .$$

Thus, $\frac{1}{b} = A + B \frac{1}{c}$, (2)

where $A = -\frac{\beta}{\alpha}$, $B = -\frac{\beta^2}{\alpha}$, or $\alpha = -\frac{B}{A^2}$, $\beta = \frac{B}{A}$. (3)

Given a set of discrete cost-benefit data, equation (2) can be used in a linear regression to find estimates for A and B. Provided there are three or more data points, estimates for A and B can be found. Using (3),

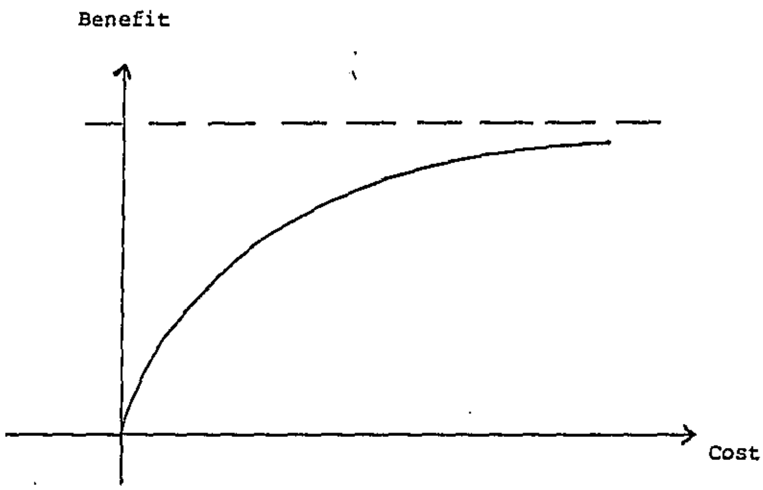


FIGURE 2. TYPICAL COST-BENEFIT CURVE

estimates for α and β can be obtained. These estimates for each product type are used in the optimization routine described in the next subsection.

From equation (1), the marginal benefit to cost ratio can be calculated and is equal to

$$\frac{db}{dc} = \frac{-\alpha}{(\beta + c)^2}$$

Using the estimates for α and β obtained above, estimates for the marginal benefit to cost ratios can be calculated. These are used repeatedly in the optimization routine.

2.2 LAGRANGIAN OPTIMIZATION ROUTINE

The Lagrangian optimization routine is the centerpiece of the Strategy Model. It is used to obtain optimal mixes of regulations in terms of regulatory cost levels at different levels of total cost and total benefit. The optimization problem can be solved using the method of Lagrange multipliers. A typical optimization problem is to find the cost for each product type such that the total benefit is maximized subject to an upper bound constraint on the total cost, and lower and upper bound constraints on the benefit and cost for each product type. First, some notations need to be introduced.

Let b_i denote the benefit for product type i,

c_i denote the regulatory cost for product type i,

α_i and β_i denote the parameters in the cost-benefit curves for product type i,

C denote the upper bound on the total cost,

b_{io} denote the lower bound on the benefit for product type i,

c_{io} denote the upper bound on the cost for product type i, and

N denote the number of product types under study.

The optimization problem described above can be stated mathematically as follows:

$$\left\{\begin{array}{lll} \text{Maximize} & \sum_{i=1}^{N} b_i = \sum_{i=1}^{N}\left(\frac{\alpha_i}{\beta_i + c_i} - \frac{\alpha_i}{\beta_i}\right) , & \\ \text{subject to} & \sum_{i=1}^{N} c_i \leq C , & \\ & b_i \geq b_{io} , & i = 1,\cdots, N \\ & c_i \leq c_{io} , & i = 1,\cdots, N . \end{array}\right.$$

Since the benefit-cost functions are all concave, the Kuhn-Tucker necessary and sufficient conditions for $\{c_i^*\}$, with the corresponding $\{b_i^*\}$, to be an optimal solution are:

$$\left\{\begin{array}{ll} C - \sum_{i=1}^{N} c_i \geq 0 & \\ b_i^* - b_{io} \geq 0 & i = 1, \cdots, N \\ c_{io} - c_i^* \geq 0 & i = 1, \cdots, N \\ \mu^* \left(C - \sum_{i-1}^{N} c_i^*\right) = 0 & \\ \sigma_i^* \left(b_i^* - b_{io}\right) = 0 & i = 1,\cdots, N \\ \nu_i^* \left(c_{io} - c_i^*\right) = 0 & i = 1,\cdots, N \\ \mu^* \geq 0 & \\ \sigma_i^* \geq 0 & i = 1,\cdots, N \\ \nu_i^* \geq 0 & i = 1,\cdots, N \\ \nabla L (\underset{\sim}{c}^*, \mu^*, \underset{\sim}{\sigma}^*, \underset{\sim}{\nu}^*) = 0, & \end{array}\right.$$

where $L(\underset{\sim}{c}, \mu, \underset{\sim}{\sigma}, \underset{\sim}{\nu}) = \sum_{i=1}^{N} b_i - \mu\left(C - \sum_{i=1}^{N} c_i\right) - \sum_{i=1}^{N} \sigma_i (b_i - b_{io})$

$$- \sum_{i-1}^{N} \nu_i \; (c_{io} - c_i),$$

and μ, $\underset{\sim}{\sigma}$, $\underset{\sim}{\nu}$ are the Lagrange multipliers.

Therefore, at optimality,

$$\frac{\partial L}{\partial c_i} (\underset{\sim}{c}_i^*) = 0 \quad i = 1, \cdots, N,$$

$$\text{i.e. } \frac{db_i}{dc_i} + \mu - \sigma_i \frac{db_i}{dc_i} + \nu_i = 0 \text{ at } c_i = c_i^*, \; i = 1, \cdots N.$$

Provided that $b_i^* > b_{io}$ and $c_i^* < c_{io}$, i.e. the constraints are not binding, then

$$\sigma_i^* = 0 \text{ and } \nu_i^* = 0$$

So

$$\frac{db_i}{dc_i} + \mu^* = 0 \text{ at } c_i = c_i^* .$$

In other words, when the upper and lower bound constraints on the costs and benefits are not binding, at optimality, all the marginal benefit to cost ratios are the same (and equal to $-\mu^*$).

This important fact is used in the optimization routine to build up a set of optimal solutions. The solution generated by the computer program is a set of optimal solutions in terms of regulatory costs for a range of total cost in fixed increments. The algorithm used in the program starts by setting all the costs to the minimum levels as constrained by the lower bounds on the benefit. At each step, an incremental cost is assigned to the product type with the maximum marginal benefit to cost ratio, provided the upper bound on cost is not yet reached. The procedure is repeated until the maximum total cost allowed is reached. Although this procedure can only produce approximate solutions, the incremental cost can be set very small and the error can be reduced to a negligible level. In the following section, the input data required to execute this optimization routine are discussed.

3. INPUT DATA REQUIREMENTS

3. INPUT DATA REQUIREMENTS

In order to execute the Lagrangian optimization routine, a set of data is required for each of the product types under study. The set consists of four elements which are the regression coefficients A and B described in Section 2, and the lower and upper bounds on the regulatory cost for each product type.

To obtain coefficients A and B, any standard linear regression routine can be used on the reciprocals of the benefit $(\frac{1}{b})$ and cost $(\frac{1}{c})$ data, as explained in Section 2. The lower and upper bounds on the regulatory cost are derived from the policy constraints, or physical constraints if there are no policy constraints. The policy constraints may set explicit bounds on the regulatory cost and these bounds form the input data directly. The policy constraints may also be in the form of bounds on the benefit. Since the cost-benefit relationship is known from equation (2) once the coefficients are known, bounds on the benefit can be translated into bounds on the cost and used as input. If there are no policy constraints on a product type, physical constraints exist that limit the minimum and maximum amounts of regulation possible. These in turn can be translated into bounds on the cost and used as input.

The format of the input data required by the computer program of the Lagrangian optimization routine is straightforward. The input data file is in card format, with each record containing the four data elements corresponding to each product type. The four data elements in each record have the format 4F7.2. The number of records in the data file is the number of product types under study.

4. OUTPUT DATA DESCRIPTION

4. OUTPUT DATA DESCRIPTION

By executing the Lagrangian optimization routine, two sets of output are generated. The first set of output consists of a table that gives the optimal mixes of regulations in terms of regulatory costs for a range of total regulatory costs and benefits. The second set of output consists of displays of benefit-cost curves graphically. The graph of total benefit vs total cost is displayed and the graph of benefit vs cost for each product (or machine) type is displayed.

The table of optimal mixes of regulations contains N + 3 columns, where N is the number of product types under study. The columns have the headings "NO.," "TOTCOST," "TOTBEN," "TYPE 01," "TYPE 02," ..., "TYPE N." NO is a simple numbering variable that counts the number of sets of optimal solutions generated. TOTCOST is the total cost of the set of regulations, and TOTBEN is the total benefit. TYPE 01 contains the optimal regulatory cost for product type 01 if the total cost is constrained to the amount given in TOTCOST or if the total benefit is constrained to the amount given in TOTBEN. TYPE 02, ..., TYPE N contain similar data. The total cost in TOTCOST increases by a small fixed increment. The total benefit in TOTBEN increases by a small and decreasing increment. Therefore, for any total cost or total benefit desired, a close approximation can be found. For each product type, using the optimal regulatory cost from the output table, the corresponding optimal regulatory level can be derived from the regulatory level to cost relationship. Since regulatory levels are usualy discrete, approximations are usually needed in deriving the optimal regulatory levels.

The second set of output contains benefit to cost curves. These curves are plotted using the IPP plotting package available on EPA's computer system. The plots are straightforward and self-explanatory. They are useful in assessing the solutions obtained from the Lagrangian optimization routine. An example of output generated by the Lagrangian optimization routine is given in Appendix B.

5. PROCEDURE FOR USING THE MODEL

5. PROCEDURE FOR USING THE MODEL [1/]

Most of the procedure for using the Strategy Model has already been explained in the previous sections. The procedure is summarized here, together with steps for operating the computer program.

1. Collect regulatory cost and benefit data corresponding to different regulatory levels or scenarios. This results in a set of discrete cost-benefit data. These data are assembled for each product type under study.

2. Use the functional form given in Section 2.1 to obtain cost-benefit curves in parametric form, with least square estimates for parameters A's and B's.

3. Select bounds on cost and benefit from policy constraints or physical constraints. Translate these bounds into bounds on regulatory cost using the estimated parametric cost-benefit function.

4. Prepare input data consisting of the estimated parameters A's and B's and bounds on regulatory cost, following the format stated in Section 3.

5. Execute the Lagrangian optimization routine which is available on EPA's computer system under the file name YAMCØ6. The input data to be used have been prepared in the previous step. The program can analyze up to 45 product (or machine) types. It is written in FORTRAN and the computation time and storage requirements are small. Any default levels should provide sufficient time and storage to execute the program successfully. A listing of the program, together with an example of an input data file is given in Appendix A.

6. Locate the solution in the output table that is closest to the total regulatory cost or total benefit desired. The table gives the optimal solution expressed in terms of regulatory cost for each product type.

7. Find the regulatory level or scenario that results in a regulatory cost that is closest to the optimal solution generated by the optimization routine. A regulatory level is found for each product type, resulting in a set of regulatory levels that give the optimal solution for a desired total cost or total benefit. This solution is only an approximation. In most applications, this approximate

1/ The computer access procedures described were in effect at the time this document was prepared. Periodically, those procedures are changed and the user should seek EPA assistance before accessing the system. JCL procedures should not change even when system access does change.

solution is adequate. However, if a greater degree of accuracy is required, or if some basic assumptions in the model do not hold (e.g. the benefits are not additive), an improved solution may be found by evaluating the regulatory options in the neighborhood of the approximate solution. If a better solution can be found by perturbation, it is likely to be the true optimal solution. The cost-benefit curves generated by the computer program may be used to illustrate the optimal solution obtained.

THE FOLLOWING IS AN EXAMPLE OF THE REQUIRED JCL TO RUN THE STRATEGY MODEL.THE DATA BASE USED TO EXECUTE THE PROGRAM IS PLACED AT THE END OF PROGRAM LISTING.

```
//EPATFP   JOB  (MUSN,RONK),RONK,PRTY=5
//S1 EXEC IPPPRCGP,PRINT=A
//FORT.SYSIN DD *
```

LOGON PROCEDURE

DIAL: 841-9560

```
PLEASE TYPE YOUR TERMINAL IDENTIFIERA
-1011-001-
PLEASE LOG IN:IBMEPA1:NCC
P 38
IBM1 IS ON LINE
WYL
ENTER WYLBUR TERMINAL TYPE
37
MODEL 37/38 TELETYPE
```

```
WYLBUR AT EPA NCC-IBM    PORT 74  THURSDAY  07/30/81  12:27:09 P.M.
07/27/81: EASYTRIEVE LOAD LIBRARIES TO BE CHANGED  SEE NEWS ALERT4
07/27/81: AIMS-2K MEETING SCHEDULED                SEE NEWS ALERT5
07/22/81: RTP/IBM USER ACCESS PROCEDURE            SEE NEWS ALERT10
07/13/81: NEW NCC-IBM USER GUIDE                   SEE NEWS ALERT3
06/19/81: NCC-IBM USERS IN WASHINGTON, DC          SEE NEWS ALERT8
USERID  ? EPAPLP
ACCOUNT ? 8888
PASSWORD? 88888888
SPECIFY GLOBAL FORMAT FOR SAVE COMMANDS
REPLY - DEFAULT, EDIT, TSO, CARD, OR PRINT
FORMAT? CARD
"LOGON" NOT FOUND IN "WYLIB"  ON USER57

COMMAND ? USE $CN.EPALYG.S2KC.WYLIB(YAMC06)
COMMAND ? RUN HOLD
1485 IS YOUR JOB NUMBER.
```

LOGOFF PROCEDURE

```
COMMAND ? LOGOFF CLR
END OF SESSION  THURSDAY  07/30/81  12:38:11 P.M.

EPAPLP/MUSN OFF WYLBUR 07/30/81 AT 12:38:10, 0.81 WUU
 0.17 CONNECT HRS., 0:00.18 TCB, 0 PAGE-SECONDS
 EXCPS: 50 DA, 0 MT, 40 TERM, 0 OTHER, 90 TOTAL
 CHARGES: $0.00 CONNECT, $0.45 WUU, $0.45 TOTAL
```

APPENDICES

APPENDIX A. PROGRAM LISTING

A listing of the computer program used for the Lagrangian optimization routine is presented in this appendix. An example of an input data file for a case with four product types is also given. The input data file is placed immediately after the program.

THIS PROGRAM IS YMCOA

```
//S1 EXEC IPPPRGGP,PRINT=A
//FORT.SYSIN DD *
C
C PROGRAM TO GENERATE TOTAL BENEFIT COST CURVE USING
C CONVEX PROGRAMMING FORMULATION AND LAGRANGIAN METHOD
C
      DIMENSION COST(21),PEN(21),AA(45),BB(45),ALPHA(45),BETA(45),
     1  CMIN(45),CMAX(45),XMBCR(45),COPT(45),TOTC(900),TOTB(900)
      REAL*8 TYPE(45)/'TYPE01','TYPE02','TYPE03','TYPE04','TYPE05',
     1          'TYPE06','TYPE07','TYPE08','TYPE09','TYPE10','TYPE11',
     2          'TYPE12','TYPE13','TYPE14','TYPE15','TYPE16','TYPE17',
     3          'TYPE18','TYPE19','TYPE20','TYPE21','TYPE22','TYPE23',
     4          'TYPE24','TYPE25','TYPE26','TYPE27','TYPE28','TYPE29',
     5          'TYPE30','TYPE31','TYPE32','TYPE33','TYPE34','TYPE35',
     6          'TYPE36','TYPE37','TYPE38','TYPE39','TYPE40','TYPE41',
     7          'TYPE42','TYPE43','TYPE44','TYPE45'/
      N=0
      M=0
      CLOW=0
      CUP=0
      TCOST=0
      TBEN=0
C READ INPUT DATA ON EACH MACHINE TYPE
  601 N=N+1
      READ(5,51,END=501) AA(N),BB(N),CMIN(N),CMAX(N)
   51 FORMAT(4F7.2)
C COMPUTE INITIAL VALUES
      BETA(N)=BB(N)/AA(N)
      ALPHA(N)=-BETA(N)/AA(N)
      XMBCR(N)=-ALPHA(N)/(BETA(N)+CMIN(N))**2
      CLOW=CLOW+CMIN(N)
      CUP=CUP+CMAX(N)
      COPT(N)=CMIN(N)
      TCOST=TCOST+CMIN(N)
      TBEN=TBEN+(ALPHA(N)/(BETA(N)+CMIN(N))-ALPHA(N)/BETA(N))
      GO TO 601
  501 CONTINUE
      N=N-1
      DELTAC=(CUP-CLOW)/(20*N)
      WRITE(6,81) (TYPE(I),I=1,N)
   81 FORMAT(' NO.   TOTCOST   TOTBEN      ',15A7/24X,15A7/26X,15A7)
      GO TO 999
C FIND MAXIMUM MARGINAL BENEFIT COST RATIO
  901 XMMAX=0
      JAY=0
      DO 701 I=1,N
      IF(XMBCR(I).GT.XMMAX.AND.COPT(I)+DELTAC.LE.CMAX(I)) JAY=I
  701 IF(XMBCR(I).GT.XMMAX.AND.COPT(I)+DELTAC.LE.CMAX(I)) XMMAX=XMBCR(I)
      IF (JAY) 959,969,801
  801 COPT(JAY)=COPT(JAY)+DELTAC
      XMBCR(JAY)=-ALPHA(JAY)/(BETA(JAY)+COPT(JAY))**2
      TCOST=TCOST+DELTAC
      TBEN=TBEN+(ALPHA(JAY)/(BETA(JAY)+COPT(JAY)))-(ALPHA(JAY)/
     1  (BETA(JAY)+COPT(JAY)-DELTAC))
      GO TO 989
  959 WRITE(6,61)
```

```
   61 FORMAT(' XYY QQQQ YYX')
      GO TO 999
C OUTPUT TOTAL COST,TOTAL BENEFIT AND INDIVIDUAL LEVELS
  989 M=M+1
      TOTC(M)=TCOST
      TOTB(M)=TBEN
      WRITE(6,71) M,TCOST,TBEN,(COPT(I),I=1,N)
   71 FORMAT('0',I3,2X,2F8.2,3X,15(F7.2)/23X,15(F7.2)/26X,15(F7.2))
      GO TO 901
C OUTPUT PLOT OF TOTAL BENEFIT VS TOTAL COST
  969 CALL PP PLOT (TOTC,TOTB,M,0.,0.,0.,0.,
     1  'QTOTAL BENEFIT VS TOTAL COSTQ','QTOTAL BENEFITQ',
     2  'QTOTAL COSTQ')
C OUTPUT PLOT OF BENEFIT VS COST OF EACH MACHINE TYPE
      I=0
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 201 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  201 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 01Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 202 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  202 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 02Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 203 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  203 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 03Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 204 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  204 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 04Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 205 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  205 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
```

```
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 05Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 206 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  206 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 06Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 207 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  207 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 07Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 208 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  208 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 08Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 209 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  209 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 09Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 210 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  210 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 10Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 211 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  211 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
```

```
    1  'QBENEFIT VS COST FOR MACHINE TYPE 110','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 212 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  212 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
    1  'QBENEFIT VS COST FOR MACHINE TYPE 120','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 213 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  213 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
    1  'QBENEFIT VS COST FOR MACHINE TYPE 130','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 214 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  214 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
    1  'QBENEFIT VS COST FOR MACHINE TYPE 140','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 215 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  215 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
    1  'QBENEFIT VS COST FOR MACHINE TYPE 150','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 216 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  216 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
    1  'QBENEFIT VS COST FOR MACHINE TYPE 160','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 217 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  217 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
    1  'QBENEFIT VS COST FOR MACHINE TYPE 170','QBENEFITQ','QCOSTQ')
```

```
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 218 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  218 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 18Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 219 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  219 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 19Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 220 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  220 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 20Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 221 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  221 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 21Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 222 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  222 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 22Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 223 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  223 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 23Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
```

```
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 224 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  224 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 240','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 225 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  225 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 250','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 226 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  226 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 260','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 227 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  227 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 270','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 228 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  228 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 280','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 229 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  229 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 290','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
```

```
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 230 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  230 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 30Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 231 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  231 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 31Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 232 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  232 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 32Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 233 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  233 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 33Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 234 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  234 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 34Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 235 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  235 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'QBENEFIT VS COST FOR MACHINE TYPE 35Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
```

```
      DC=(CMAX(I)-CMIN(I))/20
      DO 236 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  236 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 36Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 237 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  237 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 37Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 238 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  238 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 38Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 239 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  239 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 39Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 240 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  240 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 40Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 241 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  241 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1 'QBENEFIT VS COST FOR MACHINE TYPE 41Q','QBENEFITQ','QCOSTQ')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
```

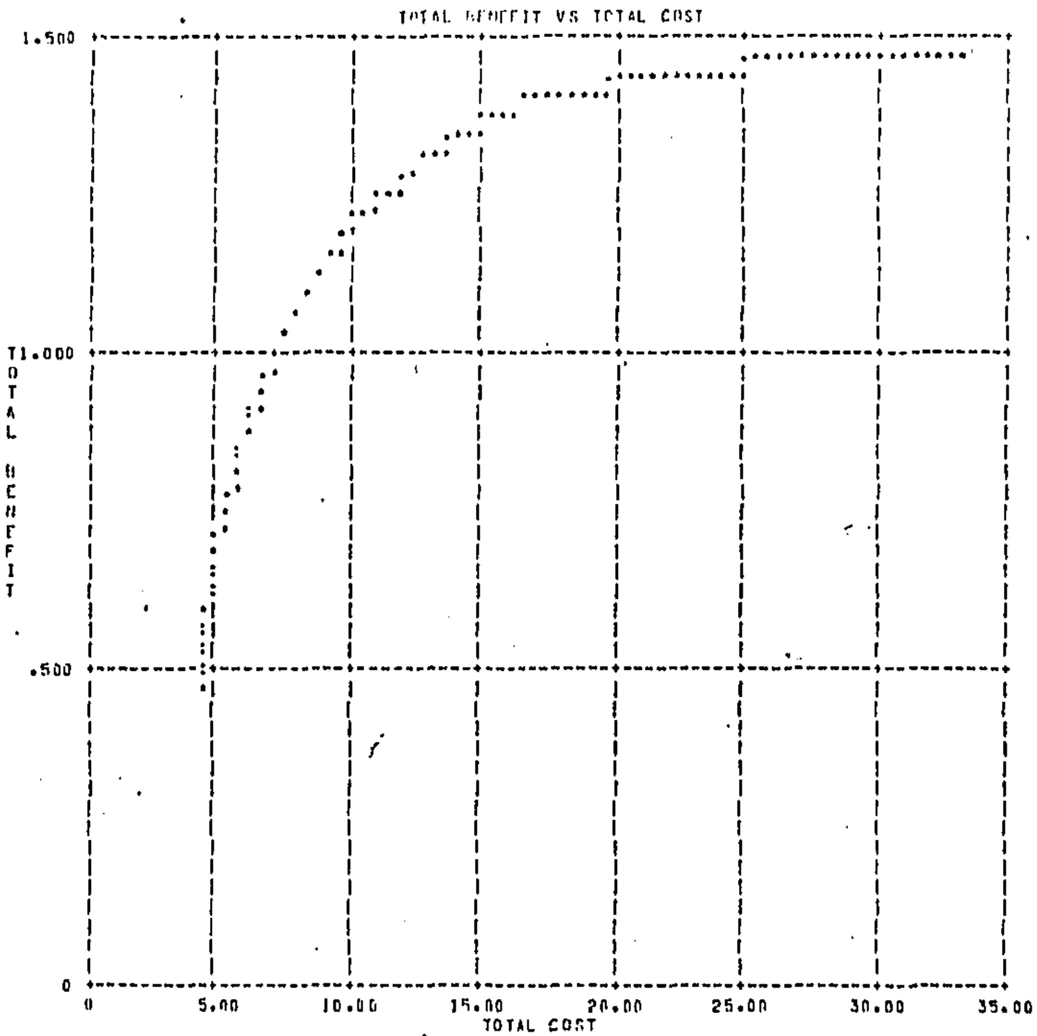
```
      DO 242 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  242 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'OBENEFIT VS COST FOR MACHINE TYPE 420','OBENEFITO','OCOSTO')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 243 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  243 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'OBENEFIT VS COST FOR MACHINE TYPE 430','OBENEFITO','OCOSTO')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 244 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  244 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'OBENEFIT VS COST FOR MACHINE TYPE 440','OBENEFITO','OCOSTO')
      IF (I.GE.N) GO TO 998
      CONTINUE
      I=I+1
      DC=(CMAX(I)-CMIN(I))/20
      DO 245 K=1,21
      COST(K)=DC*(K-1)+CMIN(I)
  245 BEN(K)=ALPHA(I)/(BETA(I)+COST(K))-ALPHA(I)/BETA(I)
      CALL PP ADVN
      CALL PP PLOT (COST,BEN,21,0.,0.,0.,0.,
     1  'OBENEFIT VS COST FOR MACHINE TYPE 450','OBENEFITO','OCOSTO')
      IF (I.GE.N) GO TO 998
      CONTINUE
  998 CALL PP CLOS
  999 STOP
      END
//GO.FT05F001 DD *
   2.81    1.15    1.00    4.00
   1.23    3.39    0.20    7.20
   2.50    1.60    0.01    15.0
   4.90    3.30    3.00    6.00
   8.50    7.80    0.00    9.30
   3.15    2.95    2.00    4.00
   4.40    3.48    0.00    2.13
   3.04    6.16    5.61    69.7
   9.75    1.24    4.51   17.30
  16.90    7.62    0.00    4.03
   3.85    1.95    2.00    4.60
   9.00    4.00    0.00    7.50
   3.00    5.50    0.00    6.00
   6.00    3.20    1.00    7.50
   2.50    3.55    0.00    4.50
   3.40    6.40    5.00    9.00
   1.32    9.75    1.32    2.55
```

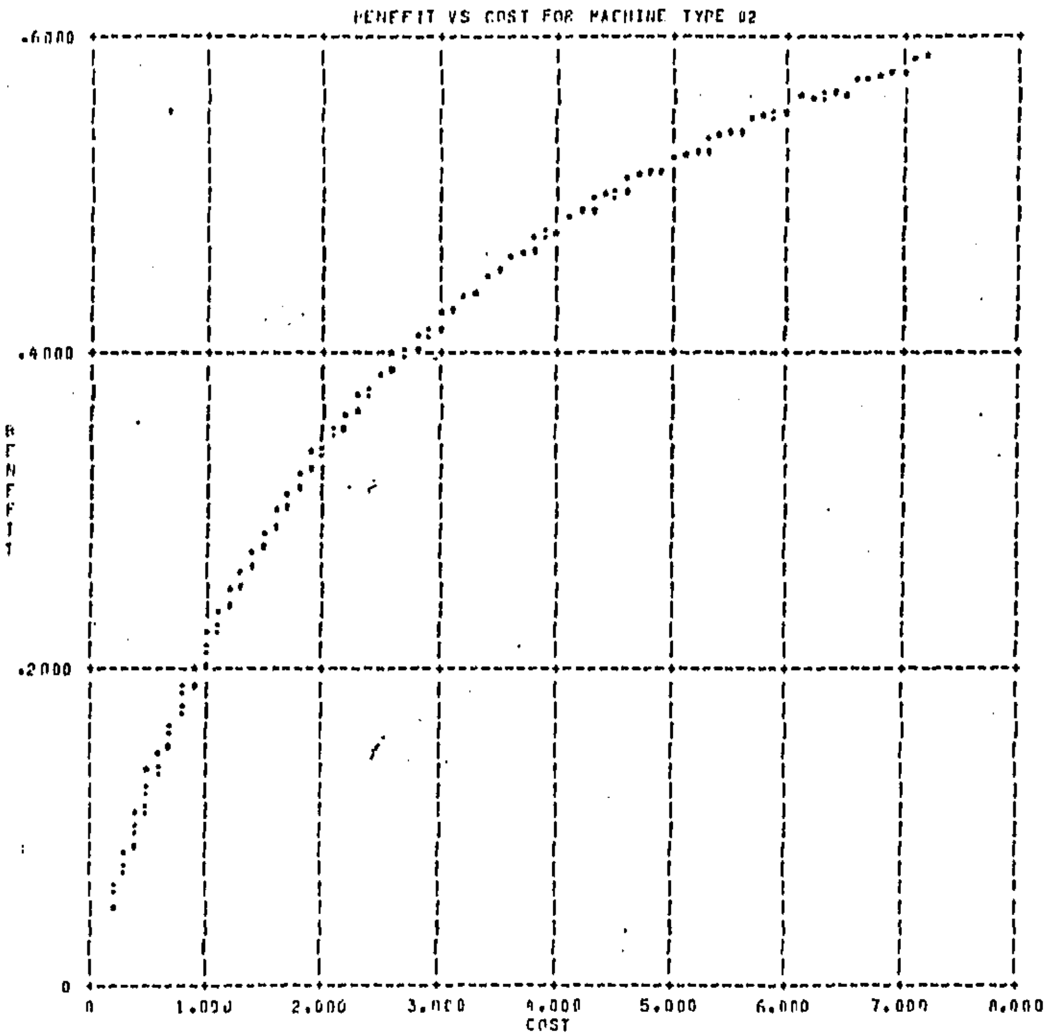
APPENDIX B. EXAMPLE OUTPUT

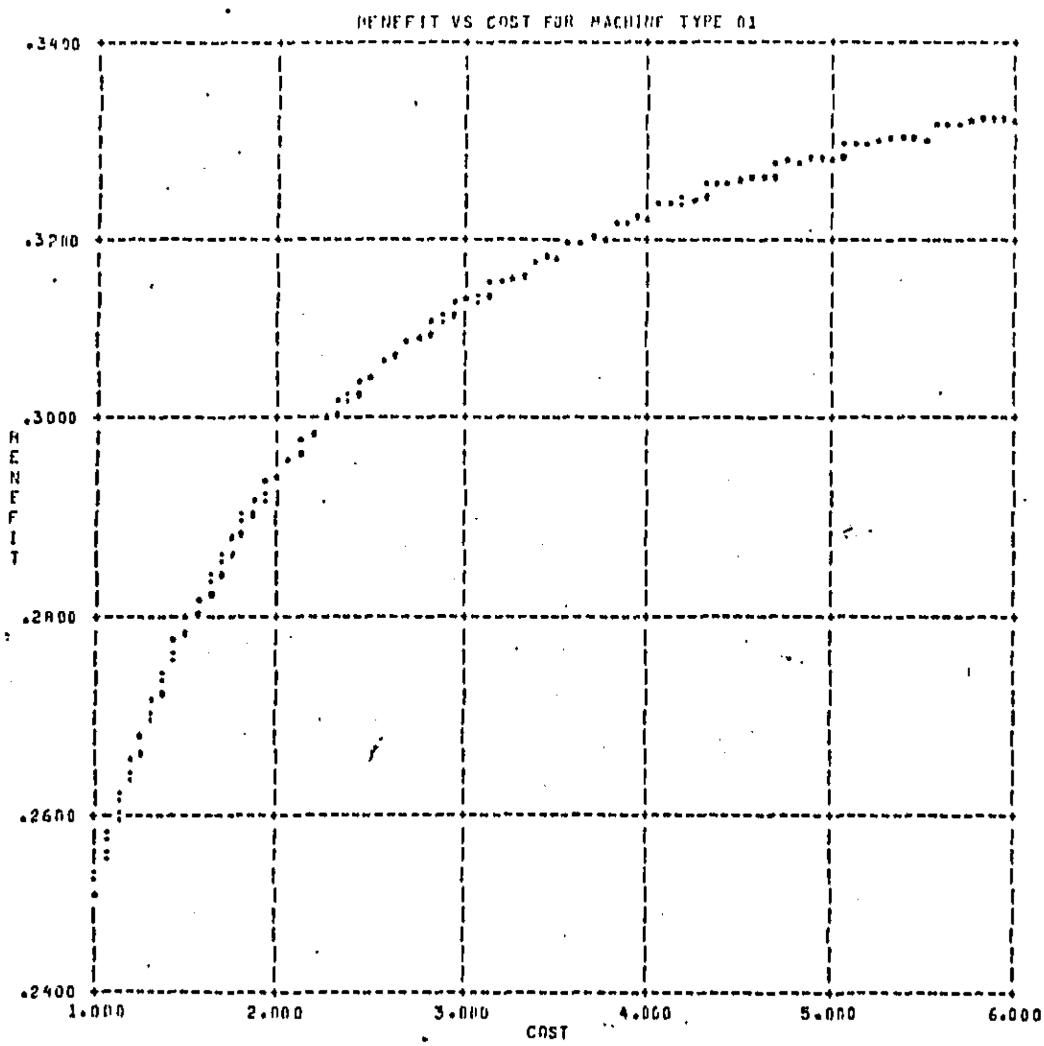
An example of output generated by the Lagrangian optimization routine using the sample input data given in Appendix A is presented in this appendix. The output consists of a table of optimal solutions followed by the graphs of total benefit vs total cost, benefit vs cost for machine types 01, 02, 03, and 04.

NO.	TOTCOST	TOTRE[illegible]	TYPE 01	TYPE 02	TYPE 03	TYPE 04
1	4.21	0.48	1.00	0.20	0.01	3.00
2	4.58	0.62	1.00	0.20	0.38	3.00
3	4.96	0.73	1.00	0.57	0.38	3.00
4	5.33	0.77	1.00	0.57	0.76	3.00
5	5.71	0.84	1.00	0.95	0.76	3.00
6	6.08	0.89	1.00	1.32	0.76	3.00
7	6.46	0.94	1.00	1.70	0.76	3.00
8	6.83	0.98	1.00	1.70	1.13	3.00
9	7.21	1.02	1.00	2.07	1.13	3.00
10	7.58	1.05	1.00	2.45	1.13	3.00
11	7.96	1.07	1.00	2.45	1.51	3.00
12	8.33	1.10	1.00	2.82	1.51	3.00
13	8.71	1.13	1.37	2.82	1.51	3.00
14	9.08	1.15	1.37	3.20	1.51	3.00
15	9.46	1.17	1.37	3.57	1.51	3.00
16	9.83	1.19	1.37	3.57	1.88	3.00
17	10.21	1.21	1.37	3.95	1.88	3.00
18	10.58	1.23	1.37	4.32	1.88	3.00
19	10.96	1.24	1.75	4.32	1.88	3.00
20	11.33	1.25	1.75	4.70	1.88	3.00
21	11.71	1.27	1.75	4.70	2.26	3.00
22	12.08	1.29	1.75	5.07	2.26	3.00
23	12.46	1.30	1.75	5.45	2.26	3.00
24	12.83	1.31	1.75	5.82	2.26	3.00
25	13.21	1.32	1.75	5.82	2.63	3.00
26	13.58	1.33	2.12	5.82	2.63	3.00
27	13.96	1.34	2.12	6.20	2.63	3.00
28	14.33	1.35	2.12	6.57	2.63	3.00
29	14.71	1.36	2.12	6.95	2.63	3.00
30	15.08	1.37	2.12	6.95	3.01	3.00
31	15.46	1.38	2.50	6.95	3.01	3.00
32	15.83	1.39	2.50	6.95	3.38	3.00
33	16.21	1.39	2.87	6.95	3.38	3.00
34	16.58	1.40	2.87	6.95	3.76	3.00
35	16.96	1.40	2.87	6.95	4.13	3.00
36	17.33	1.41	3.25	6.95	4.13	3.00
37	17.71	1.41	3.25	6.95	4.51	3.00
38	18.08	1.42	3.62	6.95	4.51	3.00
39	18.46	1.42	3.62	6.95	4.88	3.00
40	18.83	1.42	3.62	6.95	4.88	3.37
41	19.20	1.43	3.62	6.95	5.26	3.37
42	19.58	1.43	4.00	6.95	5.26	3.37
43	19.95	1.43	4.00	6.95	5.26	3.75
44	20.33	1.44	4.00	6.95	5.63	3.75
45	20.70	1.44	4.37	6.95	5.63	3.75
46	21.08	1.44	4.37	6.95	5.63	4.12
47	21.45	1.44	4.37	6.95	6.01	4.12
48	21.83	1.45	4.75	6.95	6.01	4.12
49	22.20	1.45	4.75	6.95	6.38	4.12
50	22.58	1.45	4.75	6.95	6.38	4.50
51	22.95	1.45	4.75	6.95	6.76	4.50
52	23.33	1.45	5.12	6.95	6.76	4.50
53	23.70	1.46	5.12	6.95	6.76	4.87
54	24.08	1.46	5.12	6.95	7.13	4.87
55	24.45	1.46	5.50	6.95	7.13	4.87
56	24.83	1.46	5.50	6.95	7.51	4.87
57	25.20	1.46	5.50	6.95	7.51	5.25
58	25.58	1.46	5.50	6.95	7.88	5.25
59	25.95	1.47	5.87	6.95	7.88	5.25

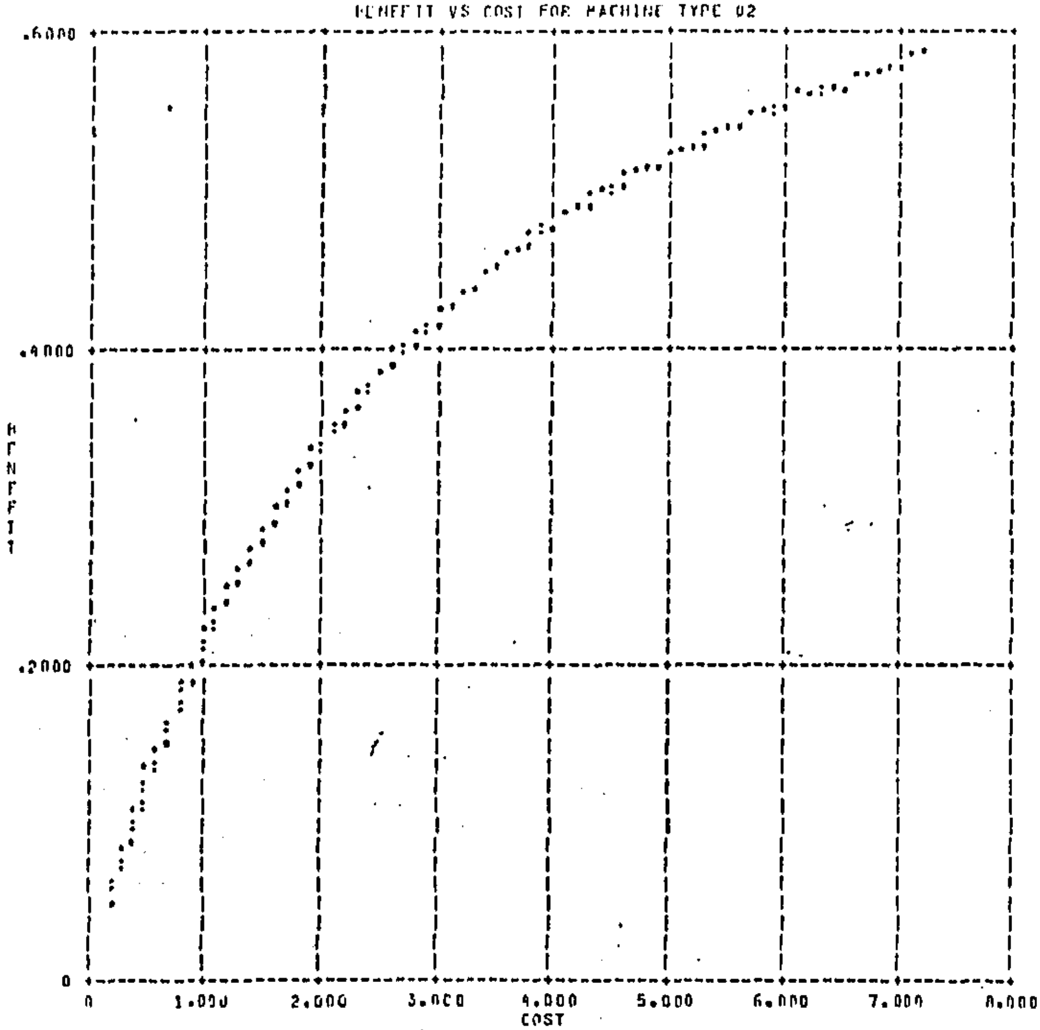
[illegible]	[illegible]	1.47	[illegible]	[illegible]	7.[illegible]	[illegible]
[illegible]	[illegible]	1.47	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	27.[illegible]	1.47	5.87	6.95	8.26	[illegible]
63	27.45	1.47	5.87	6.95	8.63	6.00
64	27.83	1.47	5.87	6.95	9.01	6.00
65	28.20	1.47	5.87	6.95	9.38	6.00
66	28.58	1.47	5.87	6.95	9.76	6.00
67	28.95	1.47	5.87	6.95	10.13	6.00
68	29.33	1.48	5.87	6.95	10.51	6.00
69	29.70	1.48	5.87	6.95	10.88	6.00
70	30.08	1.48	5.87	6.95	11.26	6.00
71	30.45	1.48	5.87	6.95	11.63	6.00
72	30.83	1.48	5.87	6.95	12.01	6.00
73	31.20	1.48	5.87	6.95	12.38	6.00
74	31.58	1.48	5.87	6.95	12.76	6.00
75	31.95	1.48	5.87	6.95	13.13	6.00
76	32.33	1.48	5.87	6.95	13.51	6.00
77	32.70	1.48	5.87	6.95	13.88	6.00
78	33.07	1.48	5.87	6.95	14.26	6.00
79	33.45	1.48	5.87	6.95	14.63	6.00

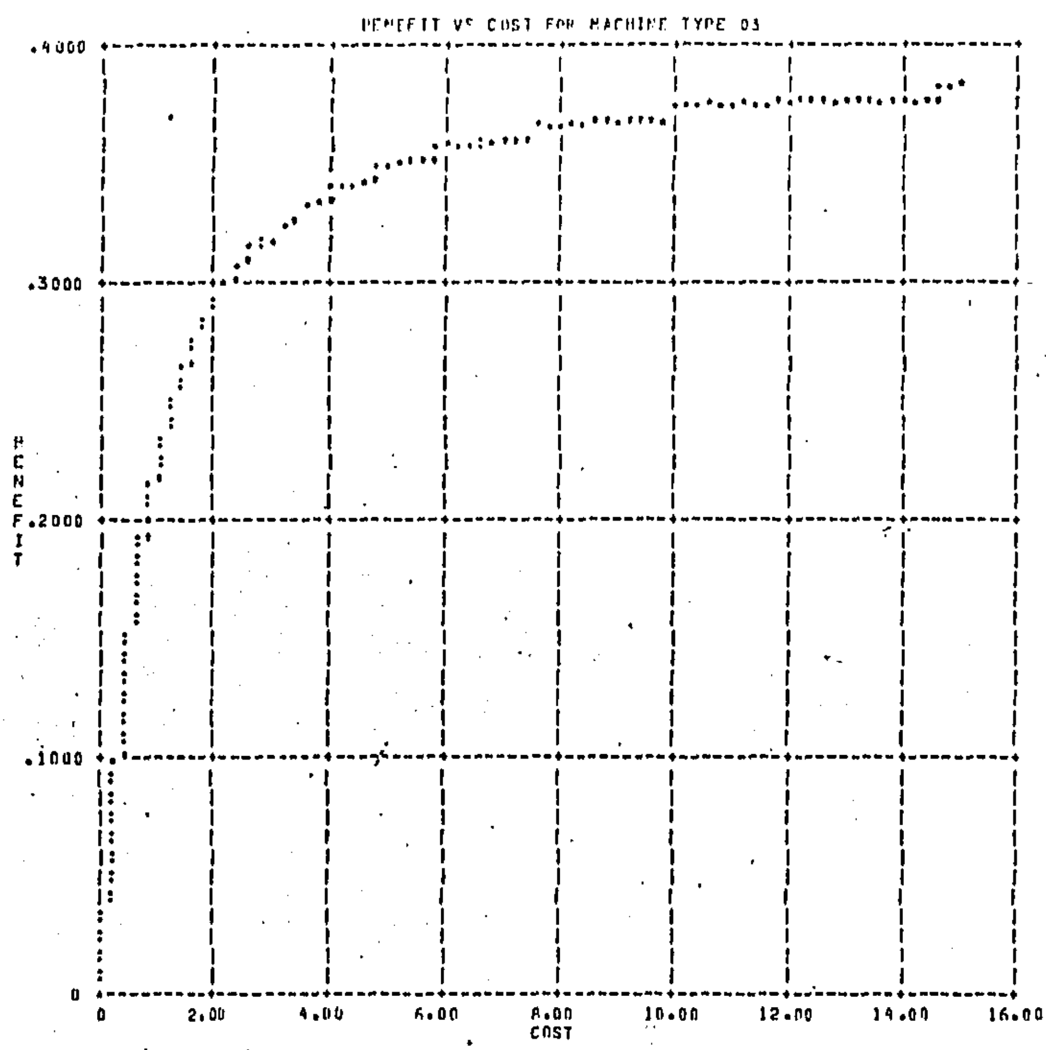

TOTAL BENEFIT VS TOTAL COST
1.500
1.000
.500
0
TOTAL BENEFIT
0
5.00
10.00
15.00
20.00
25.00
30.00
35.00
TOTAL COST


BENEFIT VS COST FOR MACHINE TYPE 02
.6000
.4000
.2000
0
BENEFIT
0
1.000
2.000
3.000
4.000
5.000
6.000
7.000
8.000
COST


BENEFIT VS COST FOR MACHINE TYPE 01
.3400
.3200
.3000
.2800
.2600
.2400
BENEFIT
1.000
2.000
3.000
4.000
5.000
6.000
COST

BENEFIT VS COST FOR MACHINE TYPE 02




BENEFIT VS COST FOR MACHINE TYPE 03
.4000
.3000
.2000
.1000
0
BENEFIT
0
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
COST

PANEL 1 FRAME 5