PROGRAMMING OPTIMAL SUGGESTIONS IN THE DESIGN CONCEPT PHASE: APPLICATION TO THE BOOTHROYD ASSEMBLY CHARTS

by

Mark Jakiela
Graduate Student

Panos Papalambros
Assistant Professor

A. Galip Ulsoy
Assistant Professor

Department of Mechanical Engineering
and Applied Mechanics
The University of Michigan
Ann Arbor

February 1984

Revised June 1984

CONTENTS

ABSTRACT

Traditional creativity-enhancement ideas and modern data handling by computing machines can be merged to create programs that make optimal suggestions to the designer. These suggestions are most useful during the design concept phase where a synthesis is attempted. The appropriate structuring of such programs is generally data- or problem-specific. From the study of specific applications general rules may emerge. This article demonstrates that approach with an application to the data contained in charts proposed by Boothroyd as guides to design for assembly.

INTRODUCTION

Among all the traditionally accepted phases in the design process, synthesis has consistently defied any attempts at automation. It is generally agreed upon that synthesis requires creativity, an elusive human ability that has not been implemented on a computing machine. In spite of eloquent arguments about a science of design [1], it appears that there will be some time before such a science can be realized, and some claim it never will. Accepting then that creative synthesis, or the *design concept phase*, will remain a human task for some time one may ask what can be done to enhance the capabilities of the designer in that phase.

The strong interest in the creative process that grew in the 1960's and early 1970's was due in part to a desire for improving human problem-solving and design synthesis by extension. Many semi-formal activities, such as morphological analysis brainstorming and synectics [2,3], were proposed in order to assist in the generation of new ideas. All these methods primarily rely on the idea that a random, spontaneous, even forced, mixing of not apparently related components has a good chance of generating a new, "creative" synthesis. Experiences with these methods show that they are useful mostly as training in a way of thinking, rather than as actual everday tools.

From the automation side, the computer-assisted design analysis is of clear importance for synthesis since many alternatives can be explored reasonably quickly and easily. However, ,it does not contribute to synthesis *per se*. The most recent focus on consultation programs or expert systems is closer to the point. A recent lay review of expert systems in mechanical engineering with bibliography can be found in [4]. However, the current format of expert systems, as developed for fields other than design, is not useful in the design concept phase because it is analysis-oriented.

A simple way to consolidate the difficulties discussed in the two approaches above, is to develop a program which can make suggestions to the designer during the conceptualization of the design. The suggestions may be based on a desired goal that the design should achieve, and expert knowledge about what design properties must be observed at the goal state. The key idea here is that, as the designer is prompted to explore the possibility of implementing a particular property, he is free, and in another sense forced, to think of different ways that this can be done. So, an effect similar to "idea triggers" in brainstorming is achieved. Furthermore, the suggestions can be made in such a way, that if they are all followed, an optimal solution will be achieved. Here, "optimal" is properly defined relative to an objective criterion which allows ranking of alternative designs. If a suggestion is not followed, the next best design will be pursued.

This approach illustrates an optimization procedure that goes beyond the traditional mathematical programming formulation. It would be useful in the early stages of design, where a configuration is sought. Then a useful mathematical model can be built for further analysis and optimization. In this latter phase the expert system format can become useful [5].

The present article illustrates the above ideas with a specific application to part of the knowledge provided by Boothroyd for evaluating designs for utomatic assembly [6]. The optimization ideas here are restricted within the scope provided by the Boothroyd data. No attempt is made to examine the adequacy of the Boothroyd evaluation method for optimal design in the general context of assembly. For example, no optimization is performed with respect to required assembly rate, which would be possibly important in a more general setting. The emphasis here is specific to the handling of the data. After a short description of the Boothroyd data as presented in the design charts, we will explore how the data can be structured to derive suggestion rules for proper interaction with the designer. The actual programming implementation will then be presented with an example. Some experiences of running the program by several users will be also discussed.

A final note should be made here about the programming language. For proper expert systems that are designed to grow and accumulate large amounts of knowledge it is appropriate to use LISP. For small-size systems that are generally static, a language like FORTRAN or Pascal is quite adequate, particularly when considering that LISP is not widely used and not yet well-developed for numerical problems. In the application presented here, Pascal was used.

THE BOOTHROYD DATA

The classification system proposed by Boothroyd and his co-workers consists mainly of a set of charts which attempt to quantify design characteristics of parts from the assembly viewpoint. The design features examined are generally of geometric nature, such as various symmetries, dimensional proportions and presence of grooves or flats. Other qualities such as rigidity, surface type (stickly or not) and complex topology (nesting or tangling when handled in bulk), are included as well. These design features are associated with numerical values assigned to four basic parameters: orienting efficiency (OE), relative feeder cost (FC), additional feeder cost (DC) and relative workhead cost (WC). The latter cost is associated with insertion procedures and means of securing the part in the sub-assembly.

The relation between design properties and numerical values for the parameters is presented by dual-entry table-charts. To use the charts, one must first assign a numerical value to a "digit" associated with a particular design feature. For example, the first digit can take values 0,1 or 2 for a rotational part with ratio of two principal dimensions L/D in the ranges (0,0.8), [0.8,1.5] and (1.5,) respectively. Assignment of values to the first three digits leads to a specification of an OE-FC pair. The additional feeder cost DC value is specified by assigning values to the fourth and fifth digits. Two more numbers, referred to here as the sixth and seventh digits (though Boothroyd does not use these terms), are needed to specify the relative insertion machine cost WC.

The assignment of numerical values to all digits gives a "digit code" which specifies values for all the above parameters. With some additional information, such as feed rates, maximum part dimension and number of simultaneous assemblies, the entire assembly cost per part is calculated based on a cost function expression.

It is evident that the optimal design according to this system is the one having maximum OE and minimum FC,DC and WC. It should be noted, however, that when maximum part dimension dictates a feed rate greater than the user required feed rate, OE is not used in the cost function expression. This is a result of under-utilization of the feeding equipment and details can be found in [6]. The typical use of the charts is an analysis of an existing design in order to evaluate it. The charts can be used for synthesis in an iterative process, i.e. create a design, rank it according to the charts, redesign and repeat. Decisions about what to change during redesign are based on familiarity with the charts and some trends implied by them. This will be discussed further in a later section.

The procedure of using the charts has been implemented on a microcomputer by Boothroyd and Dewhurst [7]. This aids the designer to iterate more efficiently on a design, but the main philosophy on the use of the data remains the same, namely as an analysis tool. In the next two sections we shall discuss how the same data can be used in the suggestive mode indicated in the introduction. A full appreciation of the discussion there will require more familiarity with the charts than what was summarized above and the interested reader is urged to consult the cited reference [6].

BINARY TREE REPRESENTATION

The most natural way to represent the data in the charts appears to be one which will aid in assigning one digit at a time. This is amplified by the apparent fact that the digit indexing corresponds to design feature generality, i.e. specification of a first digit value is based on a more general design configuration decision than what is required for the second digit, and so on. This is not strictly true however and could be misleading.

Examination of a typical chart, shows that better designs, e.g. better OE-FC pairs, are found in the upper left region of the chart, while less desirable designs are found in the lower right of the chart. So design quality generally decreases along the diagonal. It would then seem appropriate to structure the design property suggestions so that the design is placed as close as possible to the upper left portion of the chart. A natural way to effect this is to recast the row and column numbers (corresponding to digit values) in a binary tree structure.

Such a structure is shown in Fig. 1. The terminating branches on the left side of the tree correspond to regions near the upper left "optimal" region of the chart. They are accessed by affirmative response to the suggested design feature. Any branching to the right will generally suggest successively inferior designs. Note that in the lower right corner of the tree, a common node exists; therefore, a more efficient representation of a directed graph rather than a tree is used there. This is not particularly significant here because of the small size of the tree, but it could be more important in larger problems.

Once the first three digits are determined by the binary tree, an OE-FC database is accessed and the values for the parameters are determined. In a similar manner, the fourth and fifth digit determination will lead to a DC database and the sixth and seventh to a WC one.

The binary tree representation appears to be a good way to organize the program in the natural fashion of the charts organization. There is, however, a major problem. Although in general better design properties are found in the upper left portion of the charts, this is not an absolute rule. In fact, there are too many exceptions to make this method practicable. What appeared to be a good decision when assigning the first or second digit, may turn out to be a poor design when the third digit is assigned. This problem is similar to the difficulties encountered by a successive coordinatewise search in numerical optimization of nonmonotonic n-dimensional functions. In the present case, the number of possibilities is finite, so a backtracking strategy could be implemented to include better suggestions that were not introduced in the first search. In a sense, proper bookkeeping would allow checking for possible "wrong turns" after the design is created and if any were found, they could be brought to the user's attention. Then partial or full redesign would have to be initiated. This approach is plausible due to the small size of the database, but still it is psychologically unattractive because the user would have the feeling of possibly wasting a lot of thinking for solutions that would be discarded after he has completed them.

For the above reasons the binary tree structure was deemed inadequate and a different representation was sought. This is discussed in the next section.

RANKED DIRECTED GRAPH REPRESENTATION

The new structure investigated takes advantage of the fact that the state space has a relatively small number of states which can be explicitly enumerated. Recall that the first three digits specify an OE-FC pair, the fourth and fifth specify DC and the sixth and seventh specify WC. Thus we have a triplet and two couplets in the correspondence

```
(1st,2d,3d)    (OE,FC)
(4th,5th)      DC                                   (1)
(6th,7th)      WC
```

Furthermore the goal state is described by the set

```
maximum OE
minimum FC
minimum DC                                          (2)
minimum WC
```

i.e. we have multiple objectives which however are ranked in the sense that preference is given to better values for OE than FC. We assume that OE does not drop from the cost function expression (i.e. the feeders are fully utilized) in order to rank designs with respect to OE, a parameter which quantifies part design rather than economic use of feeders. If OE is dropped, the program may yield false solutions. It should be noted that there are no explicit constraints, but the designer, by responding to the suggestions, implicitly introduces or removes constraints. Thus the goal state (2) is optimal when the constraints allow for it to be reached. If this does not happen, the goal state will not be (2) but another one which can be considered as a constrained optimum, or better as a *satisficing* solution [1,8].

A characteristic of the solution space is that it has more than one element. For example, the best triplet will assign

(OE,FC) = (0.9,1.0) for rotational parts, but there are several triplets that can do that, e.g. (2,0,0), (2,1,5), (2,0,5), (2,1,0). This leads to including all triplet sets with the same quality, i.e. (OE,FC) value, into one group. The groups are then ranked according to their quality. Figure 2 shows the first two such groups for the rotational triplets. The grouping is done once manually for both rotational and non-rotational parts and also for the couplets corresponding to DC and WC values. Thus the entire database is completely structured and ranked in these groups. The groups are sequentially numbered, the first one having the best (OE,FC) values.

It is important to recognize how the values of (OE,FC) can be achieved within a group. Each group contains a list of states which are connected with OR statements. As mentioned earlier all the group states have the same (OE,FC) value. Each state now contains three design characteristics connected with AND statements. These three characteristics correspond to specific values of the first three digits. Since each state may often differ from another only by the value of one digit, most states in one group will have similar design features.

To illustrate the above ideas let us consider Group 1 in Fig. 2 and examine the program's interaction with the user. At the point when Group 1 becomes relevant, a decision has already been made that the part will be rotational. Now the program first suggests that the part have the general shape of a long cylinder, a property related to the First Digit. If the designer responds that this is possible, the program next suggests as a single statement that the part can be alpha-symmetric or slot-fed with its center of mass below supporting surfaces, properties related to the Second Digit. Finally, if the designer agrees to this, the program suggests as a <u>single</u> statement that the part be beta-symmetric <u>or</u> have a beta-asymmetric groove or flat seen in end view, properties related to the Third Digit. If the designer agrees again, a triplet value has been reached and the search stops.

The next question to be addressed is how to link the groups together. This is done by a directed graph representation where the groups are examined in hierarchical serial manner. An example of a typical suggestion structure is shown in Fig. 3. Assume that the program is considering a ranking within Group 1 by suggesting that the part be a long cylinder. If the user responds negatively, there is no reason to consider Group 2, since all the states (triplets) in that group have "long cylinder" as the First Digit property. Thus the program proceeds to consider the less desirable Group 3 having "short cylinder" as the First Digit property. Note that, while still in Group 1, if the user accepts the First Digit properties, then Group 2 will be again by-passed. Only if the Third Digit property suggestions are rejected, the program will consider Group 2.

It is evident now that Fig. 3 represents essentially an entire rearrangement of the Boothroyd chart data. The columns represent different values of the First Digit, while the rows different states of the same quality, in decreasing quality ranking. Thus, in Fig. 3 the Groups 3 and 4 have the same quality but different First Digit property, i.e. "short cylinder" versus "disk". If any of the Group 3 suggestions are rejected, consideration of Group 4 begins. If that group is also rejected, the program proceeds to Group 5, a new "long cylinder" group. Thus rejection of the "long cylinder" suggestion in Group 1 or 2 does not preclude its consideration later in the search.

At this point, an undesirable situation seems to occur because a rejected property is reintroduced. To understand why this is justified, one may first notice that in the discussion of the Group 1 above, the two Second Digit properties and the two Third Digit properties were suggested independently. This is because all four combinations of these properties are present in this best group. If they were not, after the Third Digit suggestions the program would present a suggestion of the form:

(Second Digit AND Third Digit)
OR (Second Digit AND Third Digit) (3)

in order to present all possible combinations of the group. Thus the First Digit property remains constant within a group, avoiding the need for a rather unwiedly suggestion of the form:

(First Digit AND Second Digit AND Third Digit)
OR (First Digit AND Second Digit AND Third Digit) (4)

The result is that, as the user tries to evolve and rank the design (in this first group), he has to respond to only three program suggestions. This then brings us to the issue of efficiency and appropriateness of the way the suggestions themselves are structured. The serial hierarchy described above allows eliminating consideration of a group with a minimum number of questions. Moreover, it utilizes the idea that the user should never be given the same suggestion consecutively. Repeating the same suggestion twice was considered likely to create a degree of frustration and to stifle creativity. If during a search, a rejected property is reintroduced, as in above description of Fig. 3, at a lower value, the designer can rethink the whole situation and perhaps simply restart the search with some fresh ideas. This can be done quite efficiently, at least for this size of database.

The serial hierarchy structure of the data described above is summarized as follows. The OE-FC triplets, DC and WC couplets are ranked in the ordered groups. These listings are used by the program for making suggestions to the user about the design being created. The user accepts or rejects the suggestions and synthesizes the design accordingly. When enough suggestions have been accepted, the program assigns the pertinent cost parameters from the set of values for OE,FC,DC and WC. When all parameters have been assigned, the Boothroyd cost calculations for the part design are performed and presented to the user.

PROGRAM IMPLEMENTATION

The ranked directed graph representation was coded in an interactive program written in the OMSI Pascal language on a DEC 11/34 using the RSX/11M operating system. As mentioned in the introduction, Pascal was selected as the programming language because of its wide use, good structure and the small size of the database. A sample run of the program is shown in Figures 4 to 7.

The various functions of the program are initiated through response to a menu. The digits are assigned in the sequence (1st,2d,3d), (4th,5th) and (6th,7th). Each part property suggested is preceded by an ordered pair of two integers within parentheses. The first integer refers to the Boothroyd digit place and the second integer refers to the number suggested for that place by the associated property. For example, (1,2) refers to assigning "2" as the suggested value of the First Digit. Thus an easy reference to the original Boothroyd charts is maintained. Numbers larger than nine in these parentheses reflect the fact that some of the Boothroyd Charts are divided into horizontal blocks, numbered in the charts themselves identically across the horizontal blocks. For instance, (7,19) refers to Chart No. 8 in ref. [6] and means column number 9 in the middle block. A complete description of these details is not important here. The interested reader could easily understand these by comparing the program with the actual data in [6].

In a typical design interaction, the user will sit with a sketch pad or a graphics tablet and try to visualize and sketch design configurations as prompted by the program suggestions. To avoid excessive bookkeeping in the program, the user must record and later imput the specific Boothroyd digit values corresponding to an accepted property within a group, because of the inherent ambiguity in an affirmative response to an ... OR... OR... OR... statement. Finally, the menu allows redoing many choices easily and recalculating the cost results reflecting the changes.

DISCUSSION

A first observation is that, although the entire programming effort is aimed at a synthesis aid, analysis and redesign can be effected using the program in a very straightforward and efficient manner. Note however that no concern was given to the elimination of constituent parts of an assembly, or the choice of assembly machinery, issues extensively discussed in [6]. The goal here has been demonstrating the feasibility of constructing suggestion programs and the implications with respect to how existing data is viewed, structured or manipulated.

There were several principal ideas that were adhered to for the development of the program:

1. The suggestions are made in a trully optimal way, i.e. positive response guarantees the best possible design.
2. Decisions are reached with a minimum number of questions.
3. No single suggestion is repeated consecutively.
4. Data structures developed for analysis may not be the most appropriate for design synthesis.

Some initial testing of the program with several users (mechanical engineering senior students) indicates that there are some drawbacks in the program interaction with the users. It appeared that the triplet-couplet-couplet suggestion routine was rather irritating. Many commented that they would like to assign one digit at a time, as would be done with the binary tree structure. The re-appearance of the same part suggestion, even much later than the previous time, was also considered irritating. Also, some users would answer the questions without much thinking and complete the design study without having a real design concept outlined.

This testing indications can be viewed appropriately if we mention that the users had only two hours instruction in both the Boothroyd system and the program - time clearly too limited. As the motivation behind the program was explained in some detail, the suggestion routine was accepted more rapidly. It was clearly evident, however, that the program did influence the decisions made during the design synthesis process. The negative attitudes observed may have been a result of the testing procedure rather than the program itself.

CONCLUSION

A new type of design aid has been described, where optimal suggestions are made in order to help the designer arrive at a desirable configuration. The programming needs of such a task are generally different than those of usual programming for numerical data manipulation. Since very little experience exists yet, the best approach appears to be handling the representation of specific databases first and then try to interprete the results for extraction of more general principles. This approach was followed here with the study of the Boothroyd chart data for assemblability.

ACKNOWLEDGEMENT

This research was supported by the International Business Machines Corporation for the project "Optimal Design for Automatic Assembly using Manipulators" at The University of Michigan's Center for Robotics and Integrated Manufacturing. This support is gratefully acknowledged.

REFERENCES

1. Simon, H. A., The Sciences of the Artifical, 2d. Ed., The MIT Press, Cambridge, Mass., 1981.

2. Osborn, A. F., Applied Imagination, C. Scribner's Sons, N.Y., 1963.

3. Gordon, W. J. J., Synectics, Collier Books, N.Y., 1961.

4. Dixon, J. R., and M. K. Simmons, "Computers that Design: Expert Systems for Mechanical Engineers", Comp. in Mech. Engr., ASME, N.Y., Nov. 1983, pp. 10-18.

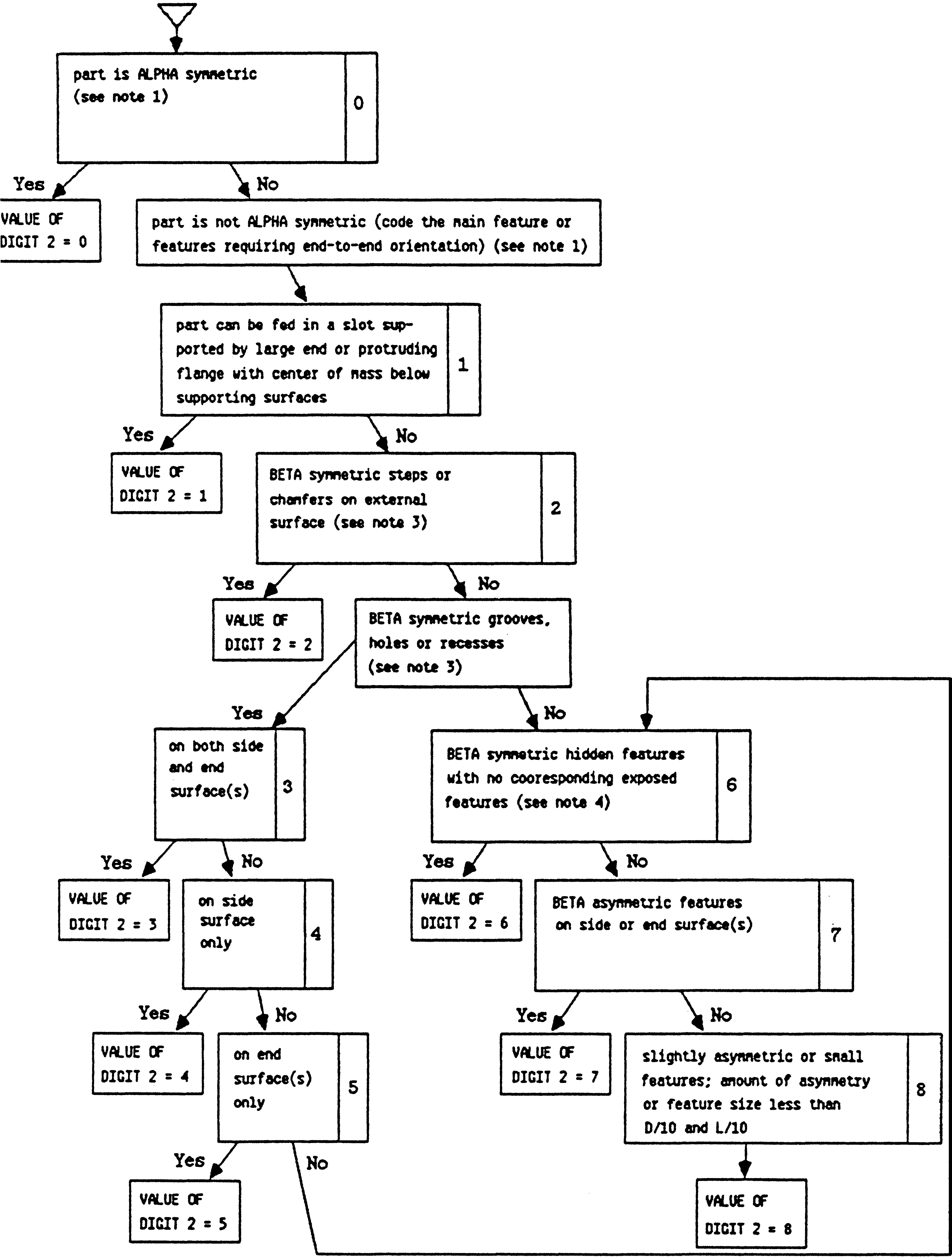
5. Azarm, S., and P. Papalambros, "A Case For A Knowledge-Based Active Set Strategy", Trans ASME, J. of Mech., Transm., and Automation in Design, Vol. 106, No. 1, March 1984, pp. 77-81.

6. Boothroyd, G., and P. Dewhurst, Design for Assembly - A Designer's Handbook, Dept. of Mech. Engr., Univ. of Mass., Amherst, Mass., 1983.

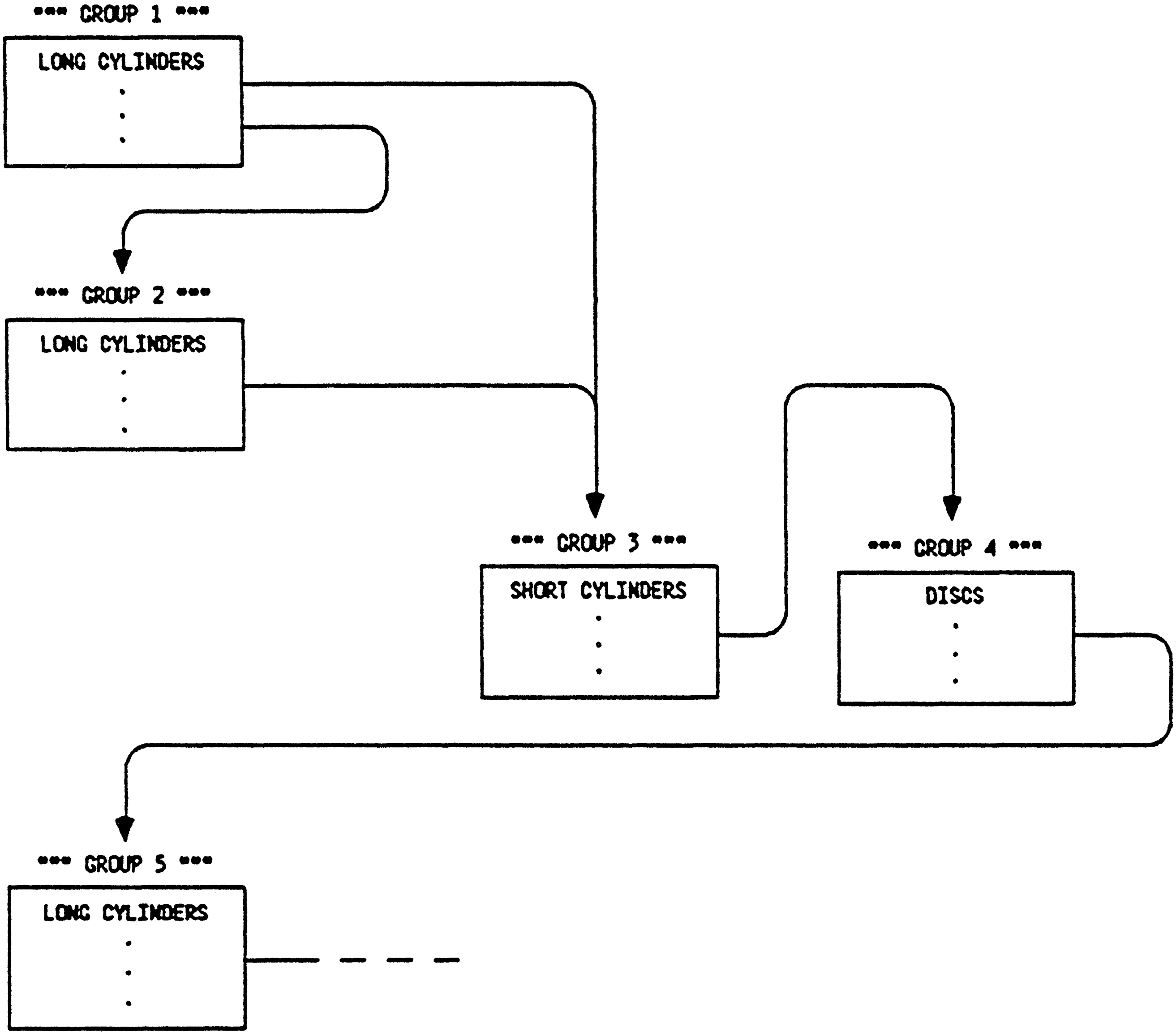
7. Boothroyd, G., and P. Dewhurst, "Computer-Aided Design for Assembly", Assembly Engineering, Feb. 1983.

8. Wilde, D. J., Globally Optimal Design, Wiley Interscience, N.Y., 1978.

LIST OF FIGURES


part is ALPHA symmetric (see note 1)
0
Yes
No
VALUE OF DIGIT 2 = 0
part is not ALPHA symmetric (code the main feature or features requiring end-to-end orientation) (see note 1)
part can be fed in a slot supported by large end or protruding flange with center of mass below supporting surfaces
1
Yes
No
VALUE OF DIGIT 2 = 1
BETA symmetric steps or chamfers on external surface (see note 3)
2
Yes
No
VALUE OF DIGIT 2 = 2
BETA symmetric grooves, holes or recesses (see note 3)
Yes
No
on both side and end surface(s)
3
BETA symmetric hidden features with no cooresponding exposed features (see note 4)
6
Yes
No
VALUE OF DIGIT 2 = 3
on side surface only
4
Yes
No
VALUE OF DIGIT 2 = 6
BETA asymmetric features on side or end surface(s)
7
Yes
No
VALUE OF DIGIT 2 = 4
on end surface(s) only
5
Yes
No
VALUE OF DIGIT 2 = 7
slightly asymmetric or small features; amount of asymmetry or feature size less than D/10 and L/10
8
Yes
No
VALUE OF DIGIT 2 = 5
VALUE OF DIGIT 2 = 8

```
*************************************************
AN ORDERED LISTING OF CONFIGURATION SUGGESTIONS
*************************************************

*** GROUP 1 ***

LONG CYLINDERS L/D > 1.5
PART IS ALPHA SYMMETRIC
PART IS SYMMETRICAL ABOUT ITS PRINCIPLE AXIS (BETA SYMMETRIC)
OE = 0.90  FC = 1.00

LONG CYLINDERS L/D > 1.5
PART CAN BE FED IN A SLOT WITH CENTER OF MASS BELOW SUPPORTING SURFACES
PART IS SYMMETRICAL ABOUT ITS PRINCIPLE AXIS (BETA SYMMETRIC)
OE = 0.90  FC = 1.00

LONG CYLINDERS L/D > 1.5
PART IS ALPHA SYMMETRIC
PART HAS BETA-ASYMMETRIC THROUGH GROOVE OR FLAT SEEN IN END VIEW
OE = 0.90  FC = 1.00

LONG CYLINDERS L/D > 1.5
PART CAN BE FED IN A SLOT WITH CENTER OF MASS BELOW SUPPORTING SURFACES
PART HAS BETA-ASYMMETRIC THROUGH GROOVE OR FLAT SEEN IN END VIEW
OE = 0.90 FC = 1.00

*** GROUP 2 ***

LONG CYLINDERS L/D > 1.5
PART IS ALPHA SYMMETRIC
PART HAS BETA-ASYMMETRIC STEPS, CHAMFERS OR PROJECTIONS ON END SURFACES ONLY
OE = 0.90  FC = 2.00

LONG CYLINDERS L/D > 1.5
PART CAN BE FED IN A SLOT WITH CENTER OF MASS BELOW SUPPORTING SURFACES
PART HAS BETA-ASYMMETRIC STEPS, CHAMFERS OR PROJECTIONS ON END SURFACES ONLY
OE = 0.90  FC = 2.00

LONG CYLINDERS L/D > 1.5
PART IS ALPHA SYMMETRIC
PART HAS BETA-ASYMMETRIC THROUGH GROOVE SEEN IN SIDE VIEW ON END SURFACE
OE = 0.90  FC = 2.00

LONG CYLINDERS L/D > 1.5
PART CAN BE FED IN A SLOT WITH CENTER OF MASS BELOW SUPPORTING SURFACES
PART HAS BETA-ASYMMETRIC THROUGH GROOVE SEEN IN SIDE VIEW ON END SURFACE
OE = 0.90  FC = 2.00

LONG CYLINDERS L/D > 1.5
PART IS ALPHA SYMMETRIC
PART HAS BETA-ASYMMETRIC THROUGH GROOVE SEEN IN SIDE VIEW ON SIDE SURFACE
OE = 0.90  FC = 2.00

LONG CYLINDERS L/D > 1.5
PART CAN BE FED IN A SLOT WITH CENTER OF MASS BELOW SUPPORTING SURFACES
PART HAS BETA-ASYMMETRIC THROUGH GROOVE SEEN IN SIDE VIEW ON SIDE SURFACE
OE = 0.90  FC = 2.00
```


*** GROUP 1 ***
LONG CYLINDERS
*** GROUP 2 ***
LONG CYLINDERS
*** GROUP 3 ***
SHORT CYLINDERS
*** GROUP 4 ***
DISCS
*** GROUP 5 ***
LONG CYLINDERS

```
RUN PROG


--- ENTER ONE OF FOLLOWING COMMANDS ---

1 = REVIEW INSTRUCTIONS
2 = ASSIGN DIGITS 1,2,3 FOR A ROTATIONAL PART
3 = ASSIGN DIGITS 1,2,3 FOR A PRISMATIC PART
4 = ASSIGN DIGITS 4,5 (SURFACE PROPERTIES)
5 = ASSIGN PARAMETER WC (INSERTION)
6 = PERFORM COST CALCULATIONS
* = END DESIGN SESSION
2

*** ASSIGN DIGITS 1,2,3 FOR ROTATIONAL PART ***


[ATTEMPTING TO ASSIGN
 OE =    0.90 FC =    1.00]

IT IS RECOMMENDED THAT ...
     (1,2) LONG CYLINDERS L/D > 1.5
IS THIS POSSIBLE ? ...
Y

IT IS RECOMMENDED THAT ...
     (2,0) ALPHA SYM.

     OR

     (2,1) NOT ALPHA SYM., SLOT FED WITH C.M. BELOW SUPPORTING S'FACES
IS THIS POSSIBLE ? ...
Y

IT IS RECOMMENDED THAT ...
     (3,0) BETA SYM.

     OR

     (3,5) BETA ASYM. THRU GROOVE OR FLAT SEEN IN END VIEW
IS THIS POSSIBLE ? ...
Y
*** OPTIMAL PARAMETERS ASSIGNED ***


*****************************
RESULTS OF SEARCH

OE =    0.90     FC =    1.00

*****************************

PLEASE INPUT THE SPECIFIC 1,2,3 BOOTHROYD DIGITS
YOU PICKED, IN THAT ORDER AS INTEGERS ...
2,0,0
```

```
--- ENTER ONE OF FOLLOWING COMMANDS ---

1 = REVIEW INSTRUCTIONS
2 = ASSIGN DIGITS 1,2,3 FOR A ROTATIONAL PART
3 = ASSIGN DIGITS 1,2,3 FOR A PRISMATIC PART
4 = ASSIGN DIGITS 4,5 (SURFACE PROPERTIES)
5 = ASSIGN PARAMETER WC (INSERTION)
6 = PERFORM COST CALCULATIONS
* = END DESIGN SESSION
 4

*** ASSIGN DIGITS 4,5 FOR PART ***



[ATTEMPTING TO ASSIGN
 PARAMETER =    0.00]

IT IS RECOMMENDED THAT ...
     (4,0) SMALL, NON-ABRASIVE, DON'T OVERLAP, NOT DELICATE, NON-FLXBLE
IS THIS POSSIBLE ? ...
Y

IT IS RECOMMENDED THAT ...
     (5,0) NOT TANGLE OR NEST, NOT LIGHT, NOT STICKY
IS THIS POSSIBLE ? ...
Y
*** OPTIMAL PARAMETERS ASSIGNED ***



*****************************
RESULTS OF SEARCH

 PARAMETER =    0.00

*****************************

PLEASE INPUT THE SPECIFIC BOOTHROYD 4,5 DIGITS
YOU PICKED, IN THAT ORDER AS INTEGERS ...
0,0
```

```
--- ENTER ONE OF FOLLOWING COMMANDS ---

1 = REVIEW INSTRUCTIONS
2 = ASSIGN DIGITS 1,2,3 FOR A ROTATIONAL PART
3 = ASSIGN DIGITS 1,2,3 FOR A PRISMATIC PART
4 = ASSIGN DIGITS 4,5 (SURFACE PROPERTIES)
5 = ASSIGN PARAMETER WC (INSERTION)
6 = PERFORM COST CALCULATIONS
* = END DESIGN SESSION
5

 *** ASSIGN DIGITS 6,7 FOR PART ***



 [ATTEMPTING TO ASSIGN
  PARAMETER =    0.80]

 IT IS RECOMMENDED THAT ...
      (6,3) FINAL SECURING, STRAIGHT INSERT., VERTICALLY ABOVE

      OR

      (6,9) SOLIDS IN PLACE, NON-SOLIDS ADDED OR PARTS MANIPULTD.
 IS THIS POSSIBLE ? ...
 Y

 IT IS RECOMMENDED THAT ...
      (7,18) SCREWING, EASY TO ALIGN & POS'N, NO RESISTANCE

      OR

      (7,22) MECH. FSTNG., NONE OR LOCAL PLASTIC DEF., SCREWING ETC.

      OR

      (7,27) NON-MECH. FSTNG., CHEMICAL FSTNG. (ADHESIVES ETC.)
 IS THIS POSSIBLE ? ...
 Y

 IT IS RECOMMENDED THAT ...
      (6,3) FINAL SECURING, STRAIGHT INSERT., VERTICALLY ABOVE
      AND
      (7,18) SCREWING, EASY TO ALIGN & POS'N, NO RESISTANCE

      OR

      (6,9) SOLIDS IN PLACE, NON-SOLIDS ADDED OR PARTS MANIPULTD.
      AND
      (7,22) MECH. FSTNG., NONE OR LOCAL PLASTIC DEF., SCREWING ETC.

      OR

      (6,9) SOLIDS IN PLACE, NON-SOLIDS ADDED OR PARTS MANIPULTD.
      AND
      (7,27) NON-MECH. FSTNG., CHEMICAL FSTNG. (ADHESIVES ETC.)
 IS THIS POSSIBLE ? ...
 Y
 *** OPTIMAL PARAMETER ASSIGNED ***


 *****************************
 RESULTS OF SEARCH

  PARAMETER =    0.80

 *****************************

 PLEASE INPUT THE SPECIFIC BOOTHROYD 6,7 DIGITS
 YOU PICKED, IN THAT ORDER AS INTEGERS ...
 3,18
```

```
 --- ENTER ONE OF FOLLOWING COMMANDS ---

 1 = REVIEW INSTRUCTIONS
 2 = ASSIGN DIGITS 1,2,3 FOR A ROTATIONAL PART
 3 = ASSIGN DIGITS 1,2,3 FOR A PRISMATIC PART
 4 = ASSIGN DIGITS 4,5 (SURFACE PROPERTIES)
 5 = ASSIGN PARAMETER WC (INSERTION)
 6 = PERFORM COST CALCULATIONS
 * = END DESIGN SESSION
 6

 PLEASE INPUT  Y =
 MAXIMUM PART DIMENSION IN MILLIMETERS ...
 25.

 PLEASE INPUT  FR =
 REQUIRED RATE OF ASSEMBLY ...
 60.

 PLEASE INPUT (AS AN INTEGER)  N =
 NUMBER OF ASSEMBLIES PERFORMED SIMULTANEOUSLY ...
 2




         *************************

          RESULTS OF DESIGN STUDY

         *************************

--- ROTATIONAL PART ---

MAXIMUM PART DIMENSION =   25.00 MM.
REQUIRED RATE OF ASSEMBLY =   60.00 PARTS/MIN
NUMBER OF SIMULTANEOUS OPERATIONS = 2

FIVE DIGIT AUTOMATIC HANDLING CODE =
     20000

     ORIENTING EFFICIENCY 'OE' =    0.90
     RELATIVE FEEDER COST  CR = FC + DC =    1.00
     MAXIMUM BASIC FEED RATE  FM =   54.00 PARTS/MIN
     DIFFICULTY RATING FOR AUTOMATIC HANDLING  DF =    1.11
     COST OF AUTOMATIC HANDLING PER PART  CF = 0.03 * DF =    0.03 CENTS

TWO DIGIT AUTOMATIC INSERTION CODE =
     38

     RELATIVE WORKHEAD COST  WC =    0.80
     DIFFICULTY RATING FOR AUTOMATIC INSERTION  DI =    0.80
     COST OF AUTOMATIC INSERTION PER PART  CI = 0.06 * DI =    0.05 CENTS

OPERATION COST =    0.16 CENTS
```