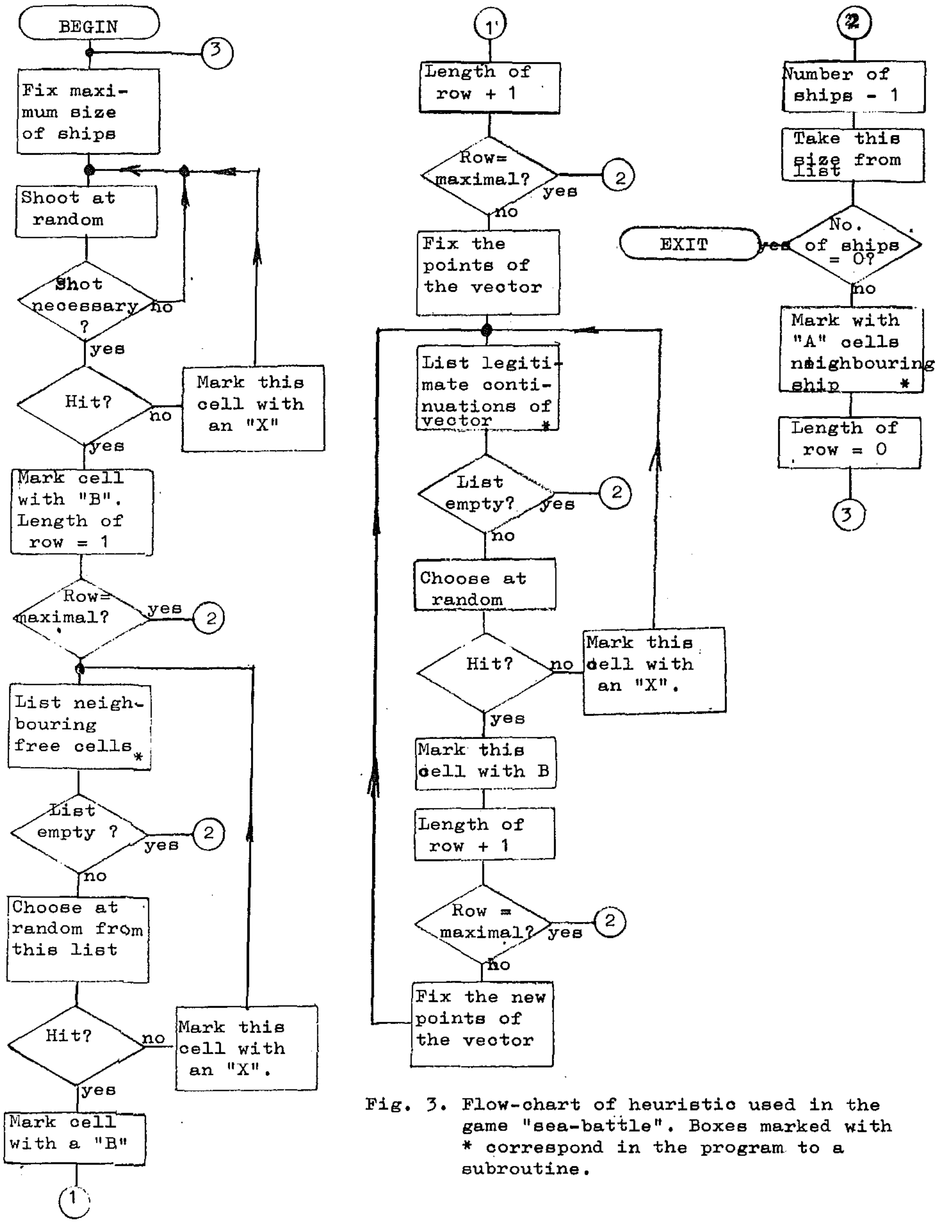

Fig. 3. Flow-chart of heuristic used in the game "sea-battle". Boxes marked with * correspond in the program to a subroutine.

We shall study the problem from the point of view pointed out in the first alternative, with the purpose of showing that this is not always successful. The heuristic that we hypothesize is the following. People will tend to shoot more or less randomly (6) until they mark a hit. Then they will try to find out whether the hit is a piece of a greater ship or not. If so they will proceed to shoot the whole ship. The search will always be directed at finding the greatest ships that still remain. Furthermore, cells neighbouring a shot ship will be avoided. This heuristic is shown as a flow-chart in figure 3. This flow-chart was transformed directly into a FORTRAN program, except for the boxes marked with an asterisk. The program has some further peculiarities for inputting and outputting information. This however, is not essential to the heuristic.

The experiment

Now we have a model (7), we want to evaluate it as capable of simulation or not. Therefore we predict that given any problem the model will do as good as human beings familiar with the game.

To check this prediction some problems were constructed at random. A FORTRAN program controlled by a random number generator was written for this purpose (8). Thus, five problems were generated. These problems were presented to five randomly selected adults (above 18 years), with as only criterion being familiar with the game. They were told how the game was to be played on this occasion and what rules were used (this because of regional differences in the rules of the game). These five persons constitute the group of subjects, as contrasted to the group "program" which consists of five trials of the program to solve these five problems. These five trials can be considered as five subjects, since each trial is distinct and completely random controlled.

For both groups the following measures were taken.

a) The number of shots needed to get the fleet to the bottom (S-score; an achievement score);
b) the number of shots that are deduced from information already given, i.e., the number of cells that are not shot because they neighbour an already shot ship (D-score; an indice of heuristic);
c) the number of blank cells minus the ones deduced (R-score); and,
d) an efficiency-index (E-score). This efficiency-index is calculated as $E = D.R/S$, because the greater the number of deductions and the greater the number of cells not used, the more efficient the heuristic, but the greater the number of shots, the less efficient the heuristic. Moreover, it should be noted that $S + D + R = 121$, the number of cells in the matrix.

The given prediction can now be stated more accurately. *First,* it is predicted that for every problem the achievement (S, R & E) will be about equal in both groups. *Second,* the same prediction holds for the indice of heuristic (D). *Third,* for both kinds of scores more variance will be attributed to the difference between problems than the difference between groups. This is predicted because differences in problems must have the same effects in both groups if the program simulates the behaviour of the subjects. *Fourth,* if both groups are equally good in solving problems, a two-way analysis of variance may reveal no significant interactions. In other words, the order of problem difficulty as shown by achievement scores and by indices of heuristic must be the same for both groups.

Results

The results of this experiment are listed in Tables 1 and 2. Table 1 shows the mean scores for each of the problems and for both groups. The means are given for S-, D-, R-, and E-scores. As can be inferred from this table, the scores for both groups are not very different. On each of the scores a two-way analysis of variance was done. The F values and their significances are listed in Table 2. The first and second predictions are confirmed by the F values of the "between groups". The third prediction is confirmed by the "between problems" F values, and the fourth by the "interactions" F values. One-way analysis of variance per problem has shown, however, that the difference in D-scores in problem 5 is statistically significant. It can be seen in Table 1 that this difference is rather great. All other analyses did not reveal significances.

So we can conclude that this model simulates human behaviour. In order to be more certain about this conclusion, additional cares were taken. We have tested whether the group of subjects was homogeneous. This was done by means of the Friedman two-way analysis of variance, a non-parametric measure (see : Siegel 1956, p. 166-172). The group appeared to be homogeneous. Furthermore, each of the subjects was asked after they solved the five problems what way they did it. All this comments correspond roughly with the described heuristic.

TABLE 1. MEAN SCORES OF SUBJECTS AND PROGRAM ON FIVE PROBLEMS.

		P1	P2	P3	P4	P5	TOTAL
SUBJECTS	S	76.2	76.	81.8	85.4	80.	79.88
	D	30.8	38.4	27.2	31.4	29.2	31.4
	R	14.	6.6	12.	4.2	11.8	9.72
	E	5.58	3.45	3.95	1.54	4.36	3.78
PROGRAM	S	77.8	80.2	84.6	81.2	77.2	80.2
	D	32.4	35.8	28.2	32.2	36.2	32.96
	R	10.8	5.	8.2	7.6	7.6	7.84
	E	4.96	2.29	2.90	2.97	3.99	3.42

TABLE 2. SUMMARY OF TWO-WAY ANALYSIS OF VARIANCE.

SOURCE	df	F-value S-score	F-value D-score	F-value R-score	F-value E-score
Within groups	40				
Problems (1)	4	2.13	6.26**	2.67*	2.2
Between groups (2)	1	.03	1.69	1.44	.25
Interaction (1)x(2)	4	.79	1.66	.79	.44
Total	49				

** p = .005

* p = .05

The conclusion is reached now. At least the subjects in our experimental group behaved in the same way as the model predicted. What does this learn us ? In fact, it does not learn us more than stated in the conclusion. The reason for this, I think, lies in the fact that we directed research at finding a heuristic used by human subjects in certain problem situations. It does not tell us anything about what humans will do in other kinds of problem situations. It is not embedded in a more general theory of problem solving, and I see no way to fit it in an existing one.

Therefore, we are forced to conclude that in some situations at least the search for heuristics cannot offer a real solution. The alternative option, then, is to orient research in simulation and artificial intelligence at testing theories and models, rather than searching heuristics. If everybody goes on exploring problems as it was the last decennium, then within a few years we shall have a lot of explored but unsolved problems (9). For this reason, at least some students in the fields discussed should try to test theories and to formalize them. For this reason too, we showed in the first section of this paper how it is possible to test theories by means of simulation.

SOME METHODOLOGICAL REQUIREMENTS

In this section some methodological requirements are listed, without much discussion, because most of them are rather straightforward.

1. It must be specified which portions of the program are *technical,* and thus not essential to the model studied. This fragments are necessary because they enable the program to run, to communicate, etc. (cf. de Groot 1969). For some fragments it is very clear whether they are technical or not. For others it can be doubted. For example, a program with a man-machine communication option that is not intended to simulate some behaviour, could by a reader of the communication be supposed to simulate. The writer should, therefore, explicitly state which parts are technical.

2. In the case of simulation as theory-testing, it should be shown that the model is *homomorphic* (or isomorphic) with the theory. This is a necessary condition; the relation must hold in order to conclude something about the theory from eventual experimental evidence.

3. Experiments ought to be replicable. Therefore it is necessary to *publish* the flow-chart and/or the program.

4. Points of comparison between model and reality should be measured in the same way, i.e., *operational definitions* should be the same. For example, time comparisons cannot be made, because the time a computer needs to perform a certain operation cannot be compared with that of human beings.

5. For the case of simulation of the game of chess, de Groot (1965) stresses that researchers should know the relevant *literature.* Simulations should show a pattern that is similar to evidence of introspective re-

search as gathered by de Groot himself. This can be generalized. Other problem regions or games are already explored, so that there too this claim can be made.

CONCLUSIONS

We have discussed some of the problems of an alternative approach to computer simulation. Some problems have deliberately been avoided. For example, we did not discuss all the possibilities of experimental designs. We avoided the problem of how to analyze the processes that lead to solutions of problems. This, however, is very important. Analysis of the processes of both program and subject must be done at a comparable level of depth. How this can be done is rather difficult to see at this moment. Some of this problems are discussed by de Groot (1966), but a solution is not given. This problem should be tackled by students of simulation or of thinking processes. It is rather astonishing that a branch of psychological investigation that was already sophisticated on introspective analysis in the turn of the century has not yet developed a clear and unequivocal coding scheme for the analysis of thinking processes. This deficiency should as soon as possible be remedied.

Another problem that was avoided is in fact, how it is practically possible to construct models, so that at once several functions of the human being such as thinking, motivation, verbal behaviour and so on, can be integrated in a computer program such that the program starts at a rather primitive level of performance and learns (or develops) to a less primitive one. This approach, which we support strongly, is very promising but faced with a great many difficulties, such as building a motivational subsystem that influences at every moment the cognitive subsystem and vice versa (cf. evidence on arousal : Berlyne 1960; Duffy 1962, Hebb 1955, Lindsley 1957, Malmo 1959). Other problems are related to ones already noticed by simulators of personality, but far from solved by now, as can be inferred from models developed by Loehlin (1963) and Abelson (1963). (More ludic trials can be found in some of the proposals in Tomkins & Messick 1963). Perhaps some help may be expected from developmental research in computer simulation.

(1) The term "heuristic" means here a simple (low-cost) solving rule that in some range of problems, where it is applicable leads readily to the solution. It is opposite to "algorithm", by which is meant is general method that always lead to the solution whatever the amount of time that is needed to reach it.

(2) Two systems are *isomorphic,* if there exists a one-one transformation if applied to the states of the one has as image the states of the other. Two systems are *homomorphic* if there exists a many-one transformation if applied to the states of the one has as image a system that is isomorphic with the other (cf. Ashby 1956, p. 86 ff.).

(3) Replication is difficult when the program is written in some specific language such as IPL or LISP, because these languages are seldom available in Europe.

(4) Designs discussed in this paper are simplified so as to show the problems involved.

(5) Both principles, that are emphatically defended by integrationist psychologists as Berlyne (1965) are of great importance in explaining some psychological events. The first of these was stressed by Hebb (1960) in his plea for a second revolution. The second is inherent to every psychological theory that has the pretension of explaining developmental events (Parmentier 1968).

(6) "Randomly" is used here rather loosely. In the case of human subjects randomness is used not in the sense of not having a "system" in shooting, but rather as not having a system that can bear a relation to the problem structure. In fact, whether the shots follow each other systematically or not, the first shot has always the chance of 20 in 121 to be a hit, and the chances for the second shot will be either 19 in 116 (after a hit) or 20 in 120 (after a miss).

(7) Note that in this example model and theory cannot be distinguished.

(8) This program and the simulation program were run on a HP 2116 B computer.

(9) This is not to say that we reject explorative investigations. To the contrary, they are necessary and are often the bringers of new ideas. However, if every student explores the problem, and nobody tries to test the ideas already found, then no progression will be made.

ABELSON R.P., *Computer simulation of "hot" cognition,* in : Tomkins S.S., & Messick S. (eds.).: Computer simulation of personality : Frontier of psychological theory, New York, Wiley, 1963, p. 277-298.

ASHBY R.W., *Design for a brain,* London, Chapman & Hall, 1952.

ASHBY R.W., *Introduction to cybernetics,* London, Chapman & Hall, 1956.

ATTNEAVE F., *Applications of information theory to psychology,* New York, 1959.

BERLYNE D.E., *Conflict, arousal and curiosity,* New York, McGraw-Hill, 1960.

BERLYNE D.E., *Structure and direction in thinking,* New York, Wiley, 1965.

BLACK F., *A deductive question-answering system* (1964), in : Minsky, M.L. (ed.) : Semantic information processing, Massachusetts, MIT, 1968.

BOBROW D.G., *Natural language input for a computer problem-solving system* (1964), in : Minsky M.L. (ed.) : Semantic information processing, Massachusetts, MIT, 1968.

COOMBS C.H., DAWES R.M. & TVERSKY A., *Mathematical psychology. An elementary introduction,* Englewood Cliffs, Prentice-Hall, 1970.

de GROOT A.D., *Thought and choice in chess,* The Hague, Mouton, 1965.

de GROOT A.D., *Perception and memory versus thought : some old ideas and recent findings,* in : Kleinmuntz B. (ed.) : Problem solving : research, method and theory, New York, Wiley, 1966, p. 19-50.

de GROOT A.D., *Methodology. Foundations of inference and research in the behavioral sciences,* The Hague, Mouton, 1969.

DE MEY M., *Paradigma's in de psychologie,* doctoraatsverhandeling (niet gepubliceerd), Rijksuniversiteit Gent, 1970.

DUFFY E., *Activation and behavior*, New York, Wiley, 1962.

EVANS T.G., *A program for the solution of a class of geometric-analogy intelligence-test questions* (1963), in : Minsky M.L. (ed.) : Semantic information processing, Massachusetts, MIT, 1968.

HEBB D.O., *Drives and the C.N.S.* (1955), in : Bindra D. & Stewart J. (eds.) : Motivation. Selected readings, Middlesex, Penguin Books, 1966, p. 63-79.

HEBB D.O., *The American revolution* (1960), in : Harper R.J.C. et. al. (eds.) : The cognitive processes. Readings, Englewood Cliffs, Prentice-Hall, 1964, p. 1-16.

HUNT E.B., *Concept learning. An information-processing problem,* New York, Wiley, 1962.

HUNT E.B., *Utilization of memory in concept learning systems,* in : Kleinmuntz B. (ed.) : Concepts and the structure of memory, New York, Wiley, 1967, p. 77-106.

HUNT E.B., *Computer simulation : Artificial intelligence studies and their relevance to psychology,* in : Annual Review of Psychology, 1968, p. 135-169.

HUNT E.B., MARIN J. & STONE P.J. : *Experiments in induction,* New York, 1966.

LINDSLEY D.B., *Psychophysiology and motivation* (1967) in : Harper R.J.C. et. al. (eds.) : The cognitive process. Readings, Englewood Cliffs, Prentice-Hall, 1964, p. 53-91.

LOEHLIN J.C., *A computer program that simulates personality,* in : Tomkins S.S. & Messick S. (eds.) Computer simulation of personality : Frontier of psychological theory, New York, 1963, p. 189-211.

MALMO R.B., *Activation : a neuropsychological dimension* (1959), in : Harper R.J.C. et. al. (eds.) : The cognitive processes. Readings, Englewood Cliffs, Prentice-Hall, 1964, p. 32-52.

McCARTHY J. et. al., *LISP 1.5 Programmer's manual*, Cambridge, Massachusetts, MIT press, 1969.

MILLER G.A., GALANTER E., PRIBRAM K.H., *Plans and the structure of behavior,* New York, Holt, 1960.

MINSKY M.L. (ed.), *Semantic information processing,* Cambridge, Massachusetts, MIT Press, 1968.

NEWELL A., SIMON H.A. & SHAW J.C., *Elements of a theory of human problem solving* (1958), in : Anderson R.C. & Ausubel D.P. (eds.) : Readings in the psychology of cognition, New York, Holt, Rinehart & Winston, 1965, p. 133-157.

NEWELL A, SHAW J.C. & SIMON H.A., *The processes of creative thinking,* in : Gruber H.E. et. al. (eds.) : Contemporary approaches to creative thinking, New York, Atherton Press, 1962, p. 63-119.

NEWELL A. & SIMON H.A. : *Computers in psychology,* in Luce R.D. et. al. (eds.) : Handbook of mathematical psychology, Vol. 1, New York, Wiley, 1963, p. 361-428.

NEWELL A. et. al. : *Information processing language-V manual,* 2nd ed., Englewood Cliffs, Prentice-Hall, 1964.

PARMENTIER R., *Recente ontwikkelingslijnen in de psychologie van de cognitieve processen,* Rijksuniversiteit Gent, Mededelingen en werkdocumenten van het Laboratorium voor experimentele, differentiël en genetische psychologie, Mededelingen en werkdocumenten 3, Werkdocument 1, 1968.

RAPHAEL B., *SIR : A computer program for semantic information retrieval,* in : Minsky, M.L. (ed.) : Semantic information processing, Cambridge, Massachusetts, MIT Press, 1968, p. 33-145 (1964).

SIEGEL S., *Nonparametric statistics for the behavioral sciences,* New York, McGraw-Hill, 1956.

TOMKINS S.S. & MESSICK S., *Computer simulation of personality : Frontier of psychological theory,* New York, Wiley, 1963.

UHR L. & VOSSLER C., *A pattern-recognition program that generates, evaluates and adjusts its own operators* (1961), in : Uhr L. (ed.) : Pattern Recognition, New York, Wiley, 1966, p. 349-364.

UHR L., VOSSLER C. & ULEMAN J., *Pattern-recognition over distortions by human subjects and by a computer simulation of a model for human form perception*, in : Journal of Experimental Psychology, 1963, 63, 3, p. 227-234.

APPENDIX

```
      PROGRAM BATLE
      DIMENSION MACOR(11,11),IBOAT(10)
      COMMON MFILD(11,11),IBUF(4),JBUF(4),IVECT(3,2)
C TECHNICAL AND COMMUNICATIVE PARTS OF THE PROGRAM
  220 WRITE(1,100)
  100 FORMAT(10X,"SEA-BATTLE")
      WRITE(1,101)
  101 FORMAT("RANDOM CONTROL")
      READ(1,*)RNC
      WRITE(1,102)
  102 FORMAT("CORRECT ANSWER")
      DO 1 L=1,11
    1 READ(1,103)(MACOR(L,K),K=1,11)
  103 FORMAT(11I1)
      IX =130B
      IB =102B
      IO =117B
C CLEAN MATRIX
      DO 2 L =1,11
      DO 2 K =1,11
```

```
    2  MFILD(L,K) =0
       NBOAT =10
       NSHOT =0
       NDED =0
       NSEMI =0
C MAKE LIST OF SHIPS
       NUM =4.
       MAX =1
       I =0
     5 DO 3 N =1, NUM
       I =I + 1
     3 IBOAT(I) =MAX
       MAX =MAX + 1
       NUM =NUM - 1
       IF(NUM) 4,4,5
C
C HEURISTIC
C
C FIX MAXIMUM SIZE OF SHIPS
     4 MAX =0
       DO 6 I =1,10
       IF(IBOAT(I)-MAX)6,6,7
     7 MAX =IBOAT(I)
     6 CONTINUE
C SHOOT AT RANDOM
    10 CALL RAND2(0,120,RNC,JX)
       L =JX/11 + 1
       K =JX-11*(L-1) + 1
C SHOT NECESSARY ?
       IF(MFILD(L,K))10,11,10
C COUNT SHOTS ***
    11 NSHOT = NSHOT + 1
```

```
C TEST : HIT ?
      IF(MACOR(L,K))12,12,13
C PASSAGE : SHOT IS A MISS
   12 MFILD(L,K) =IX
      GOTO 10
C PASSAGE : SHOT IS A HIT
   13 MFILD(L,K) =IB
      NSEMI =NSEMI + 1
C TEST : IS NSEMI MAXIMAL ?
      IF(MAX-NSEMI)8,8,9
C LIST POSSIBLE CONTINUATIONS
    9 CALL CONT(L,K,NUM)
C CONTINUATIONS POSSIBLE ?
      IF(NUM)8,8,14
   14 CALL RAND2(1,NUM,RNC,I)
      L1 =IBUF(I)
      K1 =JBUF(I)
C COUNT SHOTS ***
      NSHOT =NSHOT + 1
C TEST : HIT ?
      IF(MACOR(L1, K1))15,15,16
   15 MFILD(I
   15 MFILD(L1, K1) =IX
      GOTO 9
   16 MFILD(L1, K1) =IB
      NSEMI =NSEMI + 1
C TEST : IS NSEMI MAXIMAL ?
      IF(MAX-NSEMI)8,8,17
   17 IVECT (1,1) =L
      IVECT (1,2) =K
      IVECT (2,1) =L1
      IVECT (2,2) =K1
```

```
      IVECT (3,1) =0
C LIST VECTOR CONTINUATIONS
   21 CALL VLIST(NUM)
      IF(NUM)23,23,18
   18 CALL RAND2(1,NUM,RNC,I)
      L1 =IBUF(I)
      K1 =JBUF(I)
C COUNT SHOTS ***
      NSHOT = NSHOT + 1
C TEST : HIT ?
      IF(MACOR(L1, K1))19,19,20
   19 MFILD(L1, K1) = IX
      GOTO 21
   20 MFILD(L1, K1) = IB
      NSEMI = NSEMI + 1
C TEST : IS NSEMI MAXIMAL ?
      IF(MAX-NSEMI)23,23,22
   22 IVECT(3,1) = L1
      IVECT(3,2) = K1
      GOTO 21
C CLEAN VECTOR MATRIX
   23 DO 24 I = 1,2
      DO 24 J = 1,3
   24 IVECT(J,I) = 0
C PASSAGE INDICATES A SHIPS HAS BEEN FOUND
    8 NBOAT = NBOAT - 1
      DO 25 I = 1,10
      IF(IBOAT(I)-NSEMI)25,26,25
   26 IBOAT(I) = 0
      GOTO 27
   25 CONTINUE
```

```
C TEST : HAVE ALL SHIPS BEEN FOUND ?
   27 IF(NBOAT)29,29,28
   28 CALL FILL(NUM)
      NDED = NDED + NUM
      NSEMI = 0
      GOTO 4
C
C EXIT HEURISTIC
C
   29 WRITE(1,104)
  104 FORMAT("FOUND SOLUTION")
      DO 30 L = 1,11
      DO 30 K = 1,11
      IF(MFILD(L,K))30,31,30
   31 MFILD(L,K) = 10
   30 CONTINUE
      DO 32 L = 1,11
   32 WRITE(1,105)(MFILD(L,K,
   32 WRITE(1,105)(MFILD(L,K),K = 1,11)
  105 FORMAT(11A1)
      WRITE(1,106)NSHOT
  106 FORMAT("NUMBER OF SHOTS =   ",I3)
      WRITE(1,107)NDED
  107 FORMAT("NUMBER OF DEDUCED SHOTS =   ",I3)
      IREST = 121-(NSHOT + NDED)
      WRITE(1,108)IREST
  108 FORMAT("NUMBER OF BLANK CELLS =   ",I3)
      PAUSE 1
      GOTO 220
      END
C
C SUBROUTINES
```

```
C
      SUBROUTINE CONT(L,K,NUM)
      COMMON MFILD(11,11),IBUF(4),JBUF(4)
      NUM = 0
      INDEX = 1
      DO 1 I = 1,4
      IF(I-2)2,3,4
    2 I1 = -1
      J1 = 0
      GOTO 5
    3 I1 = 0
      J1 = 1
      GOTO 5
    4 IF(I-4)6,7
    6 J1 = -J1
      GOTO 5
    7 J1 = 0
      I1 = 1
    5 M1 = L + I1
      N1 = K + J1
      IF(M1)1,1,8
    8 IF(M1-11)9,9,1
    9 IF(N1)1,1,10
   10 IF(N1-11)11,11,1
   11 IF(MFILD(M1, N1))1,12,1
   12 NUM = NUM + 1
      IBUF(INDEX) = M1
      JBUF(INDEX) = N1
      INDEX = INDEX + 1
    1 CONTINUE
      RETURN
      END
```

```
C
      SUBROUTINE VLIST(NUM)
      DIMENSION LIST(3,2)
      COMMON MFILD(11,11), IBUF(4), JBUF(4), IVECT(3,2)
      NUM = 0
      INDEX = 1
      IF(IVECT(3,1))1,1,2
    1 N = 2
      GOTO 3
    2 N = 3
    3 MAX = 0
      MIN = 20
      NDEX = p
      NDEX = 0
      MDEX = 0
      IF(IVECT(1,1)-IVECT(2,1))4,5,4
    4 J = 1
      GOTO 6
    5 J = 2
    6 DO 7  I = 1,N
      q
      IF(IVECT(I,J)-MIN)8,9
    8 MIN = IVECT(I,J)
      NDEX = I
    9 IF(IVECT(I,J)-MAX)7,10
   10 MAX = IVECT(I,J)
      MDEX = I
    7 CONTINUE
      LIST(1,J) = IVECT(NDEX,J)
      LIST(N,J) = IVECT(MDEX,J)
      IF(J-2)11,12
```

```
11 I = 2
   GOTO 13
12 I = 1
13 DO 14 K = 1,N
14 LIST(K,I) = IVECT(K,I)
   IF(J-2)20,21
20 M1 = LIST(1,J)-1
   N1 = LIST(1,I)
   IF(M1)15,15,19
21 N1 = LIST(1,J)-1
   M1 = LIST(1,I)
   IF(N1)15,15,19
19 IF(MFILD(M1,N1))15,16,15
16 NUM = NUM + 1
   IBUF(INDEX) = M1
   JBUF(INDEX) = N1
   INDEX = INDEX + 1
15 IF(J-2)22,23
22 M1 = LIST(N,J) + 1
   N1 = LIST(N,I)
   IF(M1-11)24,24,17
23 N1 = LIST(N,J) + 1
   M1 = LIST(N,I)
   IF(N1-11)24,24,17
24 IF(MFILD(M1,N1))17,18,17
18 NUM = NUM + 1
   IBUF(INDEX) = M1
   JBUF(INDEX) = N1
17 RETURN
   END
C
   SUBROUTINE FILL(NUM)
```

```
      COMMON MFILD(11,11)
      NUM = 0
      IB = 102B
      IA = 101B
      DO 1  I = 1,11
      DO 1  J = 1,11
      I1 = I-1
      I2 = I + 1
      J1 = J-1
      J2 = J + 1
      IF(I1)2,2,3
    2 I1 = 1
    3 IF(I2-11)4,4,5
    5 I2 = 11
    4 IF(J1)6,6,7
    6 J1 = 1
    7 IF(J2-11)8,8,9
    9 J2 = 11
    8 IF(MFILD(I,J))11,1,11
   11 IF(MFILD(I,J)-IB)1,12,1
   12 DO 13 IX = I1,I2
      DO 13 JX = J1,J2
      IF(MFILD(IX,JX))13,14,13
   14 MFILD(IX,JX) = IA
      NUM = NUM + 1
   13 CONTINUE
    1 CONTINUE
      RETURN
      END
      END
```

*** Comments indicated with this asterisks indicate instructions that are not essential for the heuristic.