ALGORITHM 595
An Enumerative Algorithm for Finding Hamiltonian Circuits in a Directed Graph

SILVANO MARTELLO
University of Bologna, Italy

Categories and Subject Descriptors: G.2.2 [**Discrete Mathematics**]: Graph theory—*path and circuit problems*; G.m [**Mathematics of Computing**]: Miscellaneous—*FORTRAN*

General Terms Algorithms

Additional Key Words and Phrases Hamiltonian circuit, depth-first search

DESCRIPTION

Problem

Let $G = (V, A)$ be a *directed graph* (or *digraph*), where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of the n vertices and A is the set of the m arcs (v_i, v_j) in G. A *Hamiltonian circuit* in G is a permutation (s_i) of the vertices such that $(v_{s_i}, v_{s_{i+1}}) \in A$ for $i = 1, \ldots, n-1$ and $(v_{s_n}, v_{s_1}) \in A$.

The problem of finding one or more Hamiltonian circuits in a given graph (or alternatively of determining that the graph does not possess Hamiltonian circuits) is known to belong to the class of the *NP-complete* problems, so exact enumerative algorithms or heuristic techniques are generally used for its solution.

This paper presents a program for solving the following general problem, either exactly or heuristically:

(P) Given a digraph $G = (V, A)$ and a value $h (1 \leq h \leq +\infty)$, find h distinct Hamiltonian circuits in G or, if G does not possess h Hamiltonian circuits, find all the Hamiltonian circuits in G.

The most common cases of (P) are the extreme situations $h = 1$ (find a Hamiltonian circuit in G, if any) and $h = +\infty$ (find all the Hamiltonian circuits in G). The most efficient method for solving (P) is that proposed by Christofides [1] and Selby [3], described in [1], which is based on the enumerative scheme of

Received 13 March 1981; revised 19 August 1982; accepted 7 October 1982.
This work was supported by Consiglio Nazionale delle Ricerche, Italy.
Author's address. Istituto di Automatica, University of Bologna, 1-40136 Bologna, Italy.

Roberts and Flores [2]. The algorithm presented in this paper improves on the Christofides-Selby method by incorporating some heuristic techniques (mainly at steps 0 and 3) for determining the branches to follow in the decision tree.

It is assumed, with no loss of generality, that $(v_i, v_i) \notin A \forall v_i \in V$.

Algorithm

The algorithm is described as an exact procedure for solving problem (P) above. The following notation is used:

$$R_i = \{j \mid (v_i, v_j) \in A\};$$

$$C_j = \{i \mid (v_i, v_j) \in A\}.$$

The method is based on a depth-first branching strategy which progressively extends an elementary path $S = \{(v_{s_1}, v_{s_2}), (v_{s_2}, v_{s_3}), \ldots, (v_{s_{k-1}}, v_{s_k})\}$ (k denotes the current level of the branch decision tree and is used to save information relative to each active node of the tree) until either no further extension is possible or the path includes $n - 1$ arcs (possibly giving a Hamiltonian circuit if $(v_{s_k}, v_{s_1}) \in A$). In both cases the algorithm performs a *backtracking*, consisting in removing the last arc $v_{s_{k-1}}, v_{s_k})$ from S, followed by a *branching*, consisting in adding to S a new arc emanating from $v_{s_{k-1}}$. At each branching, a *reduction* phase removes from A all *useless arcs* (arcs that will never be included in S) and inserts *implied arcs* (arcs that will sooner or later be included in S) in a set I.

At *Step 0* a vertex v_r (*root* of S) is selected such that $|C_r| = \max_{v_i \in V}\{|C_i|\}$ (ties are broken by choosing v_r with minimum $|R_r|$); this increases the probability that, when S includes $n - 1$ arcs, a Hamiltonian circuit may be formed through arc $(v_{s_n}, v_{s_1} \equiv v_r)$. The initialization also involves setting $I = \emptyset$, $k = 1$, $s_1 = r$.

Step 1 performs a search for implied arcs by looking for nodes of in- or outdegree equal to one. Let (v_i, v_j) be any arc of A such that $R_i = \{j\}$ or $C_j = \{i\}$. For any such arc the largest path formed by (v_i, v_j) and previously implied arcs is built. Let $\bar{I}$ denote this path. If $|\bar{I}| < n - 1$, (v_i, v_j) is added to I, while all the arcs emanating from v_i or terminating at v_j and the arc from the end to the beginning vertex of $\bar{I}$ are removed from A (if this removal causes any vertices v_q to have $R_q = \emptyset$ or $C_q = \emptyset$, a backtracking (step 5) is performed). If $|\bar{I}| = n - 1$, then either a Hamiltonian circuit has been found (if the arc from the end to the beginning vertex of $\bar{I}$ exists) or a backtracking must occur. Step 1 is iterated until no further arc can be added to 1.

Step 2 determines whether an implied arc emanates from v_{s_k} (say $(v_{s_k}, v_{\bar{s}}) \in I$). If $\bar{s} \equiv r$ and $k < n$, a backtracking is performed; otherwise, S is extended by setting $k = k + 1$, $s_k = \bar{s}$ (if now $k = n$, either a Hamiltonian circuit has been found or a backtracking must occur). Step 2 is iterated until no further implied arc can be added to S.

Step 3 performs the branching phase. The next arc $(v_{s_k}, v_{\bar{s}})$ to be added to S is selected from the arcs not previously tried so that $\min\{|R_{\bar{s}}|, |C_{\bar{s}}|\}$ is a minimum (ties are broken by choosing $(v_{s_k}, v_{\bar{s}})$ with minimum $|R_{\bar{s}}| + |C_{\bar{s}}|$); in this way the most "critical" vertices are inserted first in S, increasing the probability that a Hamiltonian circuit can be obtained. If no feasible arc $(v_{s_k}, v_{\bar{s}})$ exists, a backtracking is performed. Otherwise, all the arcs emanating from v_{s_k} or terminating at $v_{\bar{s}}$, as well as arc $(v_{\bar{s}}, v_r)$, are removed from A, and S is extended by setting $k = k +$

1, $s_k = \bar{s}$. If the removal causes a vertex v_q to have $R_q = \emptyset$ or $C_q = \emptyset$, a backtracking is executed; if not, step 2 follows.

Step 4 is performed whenever a new Hamiltonian circuit is found. It only determines whether the required number h of Hamiltonian circuits has been completed: if so, the execution terminates; if not, a backtracking follows.

Step 5 performs the backtracking phase. Arc $(v_{s_{k-1}}, v_{s_k})$ is removed from S (if $k = 1$, all possibilities have been exhausted, so execution terminates). If $(v_{s_{k-1}}, v_{s_k}) \in I$ (i.e., if the arc was added to S at step 2), k is set to $k - 1$ and a new backtracking occurs. Otherwise (i.e., if the arc was added to S at step 3), all the arcs removed from A at level k are reinserted in A, all the arcs inserted in I at level k are removed from I, k is set to $k - 1$, and a new branching (step 3) is performed.

The heuristic version of the algorithm is obtained by imposing an upper bound on the number of backtrackings allowed. Only backtrackings on arcs $\notin I$ are considered, since arcs $\in I$ require no computational effort.

Program

The algorithm of the previous section was implemented in American National Standard FORTRAN as a main subroutine (**HC**) calling five subroutines (**PATH, FUPD, BUPD, IUPD,** and **RARC**). The whole package is completely self-contained and communication to it is achieved solely through the parameter list of **HC**. Entrance to the package is achieved by using the statement:

CALL HC(N, PR, AR, KW, NC, NB, S, N + 1, PR(N + 1), PC, AC, VR, VC, P, SUBR, RBUS, TOR).

Only the values of the first six parameters must be defined by the user prior to calling **HC**.

The input digraph $G = (V, A)$ is stored as an adjacency list. Array **AR** contains the elements of R_1 in the first $|R_1|$ locations, the elements of R_2 in the following $|R_2|$ locations, and so on. The n records of **AR** are pointed by the $n + 1$ elements of array **PR** (**PR**$(i) = \sum_{j=1}^{i-1} |R_j|$; **PR**$(1) = 0$), so that the elements of R_i are stored in locations **PR**$(i) + 1$ to **PR**$(i + 1)$ of **AR**. It follows that the value m is stored in **PR**$(n + 1)$.

The other input parameters are

N number of vertices (n);

KW output unit number on which to write the permutations (s_i) corresponding to the Hamiltonian circuits found (**KW** = −1 if no writing is desired). The permutations are written according to Format 20I5.

Two input-output parameters are used:

NC (input) upper bound on the number of Hamiltonian circuits to be found (**NC** = −1 if all are to be found);

NC (output) number of Hamiltonian circuits found;

NB (input) −1 if exact execution is required; upper bound on the number of backtrackings if heuristic execution is required;

NB (output) number of backtrackings performed (for heuristic executions, if **NB**(output) < **NB**(input), the result obtained is exact).

Table I. Search for All the Hamiltonian Circuits

	Vertex degrees in range 1–3			Vertex degrees in range 2–3			Vertex degrees in range 2–4		
n	Average (max) running time	Average (max) number of backtrackings	Average (max) number of Hamiltonian circuits	Average (max) running time	Average (max) number of backtrackings	Average (max) number of Hamiltonian circuits	Average (max) running time	Average (max) number of backtrackings	Average (max) number of Hamiltonian circuits
5	0.002 (0.004)	0.4 (2)	0.8 (2)	0.010 (0.014)	4.9 (7)	3.6 (5)	0.011 (0.021)	5.2 (10)	3.5 (7)
10	0 007 (0.034)	1.9 (12)	1.1 (6)	0.023 (0.050)	7.9 (18)	2.8 (6)	0.063 (0.141)	25.1 (54)	9.4 (20)
15	0.004 (0.012)	0.2 (2)	0.2 (2)	0.091 (0.246)	27.1 (79)	8.0 (22)	0.346 (0 912)	111.4 (309)	33.3 (110)
20	0.003 (0.007)	0.0 (0)	0.0 (0)	0.192 (0 530)	51.8 (166)	12.2 (38)	0.875 (1.912)	229.1 (512)	72.0 (157)
25	0.005 (0.014)	0.0 (0)	0.0 (0)	0 201 (0.448)	45.8 (121)	10.8 (34)	2.169 (8.722)	514.0 (2097)	129.7 (584)
30	0.004 (0.006)	0.0 (0)	0 0 (0)	0.660 (1.639)	147.0 (417)	29.5 (123)	11.219 (35.021)	2515.0 (8300)	638.3 (2462)

Note Ten problems for each entry. Times in CDC—6600 seconds.

Table II. Search for a Single Hamiltonian Circuit

	Vertex degrees in range 1–3			Vertex degrees in range 2–3			Vertex degrees in range 2–4		
n	Average (max) running time	Average (max) number of backtrackings	Average number of Hamiltonian circuits	Average (max) running time	Average (max) number of backtrackings	Average number of Hamiltonian circuits	Average (max) running time	Average (max) number of backtrackings	Average number of Hamiltonian circuits
50	0.006 (0.010)	0.0 (0)	0.0	0.102 (0.326)	7.0 (33)	0.9	0.095 (0.268)	7.9 (41)	1.0
100	0.011 (0.016)	0.0 (0)	0.0	0.344 (1.126)	14.2 (62)	1.0	0.601 (4.031)	44.1 (356)	1.0
150	0.012 (0.021)	0.0 (0)	0.0	0.525 (1.748)	13 4 (58)	1.0	1.480 (4.549)	78.0 (293)	1.0
200	0.017 (0.021)	0.0 (0)	0.0	1.068 (3.781)	27.5 (119)	1.0	1.356 (6.559)	43.4 (267)	1.0
250	0.019 (0 024)	0.0 (0)	0.0	2.255 (10.954)	51 9 (261)	1.0	6.353 (45 057)	182.2 (1344)	1.0

Note: Ten problems for each entry. Times in CDC—6600 seconds.

The only output parameter is

S(i) ith element of the permutation corresponding to the last Hamiltonian circuit found (s_i).

N + 1 and **PR**(**N** + 1) are used for adjustable dimensions. **PC**, **AC**, **VR**, **VC**, **P**, **SUBR**, **RBUS**, and **TOR** are work arrays. In the calling program, the user must dimension **PR** and **PC** at least at n + 1, **AR** and **AC** at least at m, and **S**, **VR**, **VCP**, **SUBR**, **RBUS**, and **TOR** at least at n.

All the parameters are integer. After execution, the order of the elements within the records of **AR** may be altered.

Computational Results

The code was computationally tested on randomly generated digraphs with both the indegree and the outdegree of each vertex lying in prefixed ranges. For each range and for different values of the number of nodes, 10 graphs were generated and solved on a CDC-6600.

Table I shows the results obtained by using the code for finding all the Hamiltonian circuits in small-size digraphs. The case of vertex degrees in range 1–3 shows that the algorithm is very fast in solving problems where very few (or no) Hamiltonian circuits exist. The case of vertex degrees in range 2–3 shows that the ratio (average running time)/(average number of Hamiltonian circuits) grows about linearly with n. The same growing rate is presented by the case of vertex degrees in range 2–4; in this case, however, the running times tend to be impractical because of the high number of Hamiltonian circuits.

Table II refers to use of the code for finding a single Hamiltonian circuit in large-size digraphs. It is confirmed that the algorithm easily solves cases where no Hamiltonian circuit exists (vertex degrees in range 1–3). The cases of vertex degrees in ranges 2–3 and 2–4 show that the growing rate with n is much higher for dense digraphs.

Used as a heuristic on the same problems as Table II, the code obviously gave an exact solution for all the digraphs with vertex degrees in range 1–3, for any prefixed upper bound **NB** on the number of backtrackings allowed. In the case of vertex degrees in range 2–3, the problems solved exactly were 62 percent for **NB** = 10, 78 percent for **NB** = 20, 85 percent for **NB** = 40; for range 2–4, such percentages were 50, 58, and 72 percent, respectively.

ACKNOWLEDGMENT

I am grateful to Prof. P. Toth for his helpful comments.

REFERENCES

1. CHRISTOFIDES, N. *Graph Theory—An Algorithmic Approach.* Academic Press, New York, 1975.
2. ROBERTS, S.M., AND FLORES, B. Systematic generation of Hamiltonian circuits. *Commun ACM 9*, 9(Sept. 1966), 690–694.
3. SELBY, G.R. The use of topological methods in computer-aided circuit layout. Ph.D. dissertation, London Univ., London, England, 1970.

ALGORITHM

[A part of the listing is printed here. The complete listing is available from the ACM Algorithms Distribution Service (see page 141 for order form).]

```
C SAMPLE DRIVER PROGRAM FOR HC.                                          MAN   10
C                                                                        MAN   20
C THIS PROGRAM FINDS ONE OR MORE HAMILTONIAN CIRCUITS IN A               MAN   30
C DIRECTED GRAPH OF  N  VERTICES AND  M  ARCS.                           MAN   40
C                                                                        MAN   50
C COMPLETE DETAILS OF THE PARAMETERS MAY BE FOUND IN THE                 MAN   60
C DOCUMENTATION OF SUBROUTINE HC.                                        MAN   70
C                                                                        MAN   80
C THE INPUT UNIT NUMBER IS ASSUMED TO BE  5 .                            MAN   90
C THE OUTPUT UNIT NUMBER IS ASSUMED TO BE  6 .                           MAN  100
C THE ARRAYS ARE CURRENTLY DIMENSIONED TO ALLOW PROBLEMS FOR             MAN  110
C WHICH  N .LE. 250  AND  M .LE. 2000 .                                  MAN  120
C                                                                        MAN  130
C THE PROGRAM MAY BE TESTED ON THE FOLLOWING DATA                        MAN  140
C                                                                        MAN  150
C N  =     6                                                             MAN  160
C PR =     0    3    5    8   11   13   16                              MAN  170
C AR =     3    5    6    6    3    6    1    4    5    1               MAN  180
C          2    2    3    4    2    5                                   MAN  190
C                                                                        MAN  200
      INTEGER PR(251), PC(251), AR(2000), AC(2000), S(250), VR(250),     MAN  210
     * VC(250), P(250), SUBR(250), RBUS(250), TOR(250)                   MAN  220
      NIN = 5                                                            MAN  230
      NOUT = 6                                                           MAN  240
C INPUT DATA                                                             MAN  250
      READ (NIN,99999) N                                                 MAN  260
      NP1 = N + 1                                                        MAN  270
      READ (NIN,99999) (PR(J),J=1,NP1)                                   MAN  280
      M = PR(NP1)                                                        MAN  290
      READ (NIN,99999) (AR(J),J=1,M)                                     MAN  300
      WRITE (NOUT,99998)                                                 MAN  310
      WRITE (NOUT,99997) N                                               MAN  320
      WRITE (NOUT,99996) (PR(J),J=1,NP1)                                 MAN  330
      WRITE (NOUT,99995) (AR(J),J=1,M)                                   MAN  340
      WRITE (NOUT,99994)                                                 MAN  350
C CALL HC AS EXACT PROCEDURE TO FIND (AND PRINT) A SINGLE                MAN  360
C HAMILTONIAN CIRCUIT, IF ONE EXISTS.                                    MAN  370
      NC = 1                                                             MAN  380
      NB = -1                                                            MAN  390
      CALL HC(N, PR, AR, NOUT, NC, NB, S, N+1, PR(N+1), PC, AC, VR, VC,  MAN  400
     * P, SUBR, RBUS, TOR)                                              MAN  410
      WRITE (NOUT,99993) NC, NB                                          MAN  420
C CALL HC AS EXACT PROCEDURE TO FIND (AND PRINT) ALL THE                 MAN  430
C HAMILTONIAN CIRCUITS.                                                  MAN  440
      NC = -1                                                            MAN  450
      NB = -1                                                            MAN  460
      CALL HC(N, PR, AR, NOUT, NC, NB, S, N+1, PR(N+1), PC, AC, VR, VC,  MAN  470
     * P, SUBR, RBUS, TOR)                                              MAN  480
      WRITE (NOUT,99993) NC, NB                                          MAN  490
C CALL HC AS HEURISTIC PROCEDURE TO FIND (AND PRINT) A                   MAN  500
C SINGLE HAMILTONIAN CIRCUIT, IF ONE EXISTS, WITHOUT                     MAN  510
C PERFORMING MORE THAN  2  BACKTRACKINGS.                                MAN  520
      NC = 1                                                             MAN  530
      NB = 2                                                             MAN  540
      CALL HC(N, PR, AR, NOUT, NC, NB, S, N+1, PR(N+1), PC, AC, VR, VC,  MAN  550
     * P, SUBR, RBUS, TOR)                                              MAN  560
      WRITE (NOUT,99993) NC, NB                                          MAN  570
C CALL HC AS HEURISTIC PROCEDURE TO FIND (AND PRINT) A                   MAN  580
C SINGLE HAMILTONIAN CIRCUIT, IF ONE EXISTS, WITHOUT                     MAN  590
C PERFORMING MORE THAN  4  BACKTRACKINGS.                                MAN  600
      NC = 1                                                             MAN  610
      NB = 4                                                             MAN  620
      CALL HC(N, PR, AR, NOUT, NC, NB, S, N+1, PR(N+1), PC, AC, VR, VC,  MAN  630
     * P, SUBR, RBUS, TOR)                                              MAN  640
      WRITE (NOUT,99993) NC, NB                                          MAN  650
C CALL HC AS HEURISTIC PROCEDURE TO FIND (WITHOUT PRINTING)              MAN  660
C AT MOST  2  HAMILTONIAN CIRCUITS, WITHOUT PERFORMING MORE              MAN  670
C THAN  5  BACKTRACKINGS.                                                MAN  680
      NC = 2                                                             MAN  690
      NB = 5                                                             MAN  700
      CALL HC(N, PR, AR, -1, NC, NB, S, N+1, PR(N+1), PC, AC, VR, VC,    MAN  710
     * P, SUBR, RBUS, TOR)                                              MAN  720
      WRITE (NOUT,99993) NC, NB                                          MAN  730
      IF (NC.EQ.0) STOP                                                  MAN  740
C PRINT THE LAST HAMILTONIAN CIRCUIT FOUND                               MAN  750
      WRITE (NOUT,99992) (S(J),J=1,N)                                    MAN  760
```

```
      STOP                                                              MAN  770
99999 FORMAT (10I5)                                                     MAN  780
99998 FORMAT (1H1/////////)                                             MAN  790
99997 FORMAT (6H N  = , I5)                                             MAN  800
99996 FORMAT (6H PR = , 25I5)                                           MAN  810
99995 FORMAT (6H AR = , 25I5)                                           MAN  820
99994 FORMAT (//////)                                                   MAN  830
99993 FORMAT (/I5, 10H CIRCUITS , I5, 14H BACKTRACKINGS///)             MAN  840
99992 FORMAT (4X, 4HS = , 25I5)                                         MAN  850
      END                                                               MAN  860
      SUBROUTINE HC(N, PR, AR, KW, NC, NB, S, NP1, M, PC, AC, VR, VC,   HC   10
     * P, SUBR, RBUS, TOR)                                             HC   20
C                                                                       HC   30
C SUBROUTINE TO FIND ONE OR MORE HAMILTONIAN CIRCUITS IN A              HC   40
C DIRECTED GRAPH OF  N  VERTICES ( N .GT. 1 ) REPRESENTED               HC   50
C BY THE INTEGERS  1, 2, ..., N  AND  M  ARCS.                          HC   60
C                                                                       HC   70
C HC IS BASED ON AN ENUMERATIVE ALGORITHM AND CAN BE USED               HC   80
C EITHER AS AN EXACT PROCEDURE OR AS A HEURISTIC PROCEDURE              HC   90
C (BY LIMITING THE NUMBER OF BACKTRACKINGS ALLOWED).                    HC  100
C                                                                       HC  110
C ENTRANCE TO HC IS ACHIEVED BY USING THE STATEMENT                     HC  120
C     CALL HC(N,PR,AR,KW,NC,NB,S,N+1,PR(N+1),PC,AC,VR,VC,               HC  130
C    *        P,SUBR,RBUS,TOR)                                         HC  140
C                                                                       HC  150
C THE VALUES OF THE FIRST SIX PARAMETERS MUST BE DEFINED                HC  160
C BY THE USER PRIOR TO CALLING HC. HC NEEDS  2  ARRAYS ( PR             HC  170
C AND  PC ) OF LENGTH  N + 1 ,  2  ARRAYS ( AR  AND  AC )              HC  180
C OF LENGTH  M  AND  7  ARRAYS ( S ,  VR ,  VC ,  SUBR ,               HC  190
C RBUS  AND  TOR ) OF LENGTH  N . THESE ARRAYS MUST BE                  HC  200
C DIMENSIONED BY THE USER IN THE CALLING PROGRAM.                       HC  210
C                                                                       HC  220
C HC CALLS  5  SUBROUTINES: PATH, FUPD, BUPD, IUPD, RARC.               HC  230
C THESE SUBROUTINES ARE COMPLETELY LOCAL, I.E. THE INFORMA-             HC  240
C TION THEY NEED IS PASSED THROUGH THE PARAMETER LIST.                  HC  250
C THE WHOLE PACKAGE IS COMPLETELY SELF CONTAINED AND COMMU-             HC  260
C NICATION TO IT IS ACHIEVED SOLELY THROUGH THE PARAMETER               HC  270
C LIST OF HC. NO MACHINE DEPENDENT CONSTANTS ARE USED.                  HC  280
C THE PACKAGE IS WRITTEN IN AMERICAN NATIONAL STANDARD                  HC  290
C FORTRAN AND IS ACCEPTED BY THE  FTN(EL=A)  COMPILER OF THE            HC  300
C CDC CYBER 76 (OPTION  EL=A  CHECKS PROGRAM AND SUBROUTINES            HC  310
C FOR ADHERENCE TO ANSI) AS WELL AS BY THE PFORT VERIFIER.              HC  320
C THE PACKAGE HAS BEEN TESTED ON A  CDC CYBER 76 , ON A  CDC            HC  330
C 6600 , ON AN  IBM 370/158  AND ON A  DIGITAL VAX 11/780 .             HC  340
C                                                                       HC  350
C MEANING OF THE INPUT PARAMETERS:                                      HC  360
C N      = NUMBER OF VERTICES.                                          HC  370
C PR(I) = SUM OF THE OUT-DEGREES OF VERTICES  1, ..., I-1               HC  380
C         ( PR(1) = 0 ,  PR(N+1) = M ).                                 HC  390
C AR     = ADJACENCY LIST. THE ELEMENTS FROM  AR(PR(I)+1)  TO           HC  400
C          AR(PR(I+1))  ARE A RECORD CONTAINING,IN ANY ORDER,           HC  410
C          ALL THE VERTICES  J  SUCH THAT ARC  (I,J)  EXISTS.           HC  420
C          THE GRAPH SHOULD NOT CONTAIN ARCS STARTING AND               HC  430
C          ENDING AT THE SAME VERTEX.                                   HC  440
C KW     = UNIT TAPE ON WHICH TO WRITE THE HAMILTONIAN CIR-              HC  450
C          CUITS FOUND, ACCORDING TO FORMAT  20I5 .  KW = - 1           HC  460
C          IF NO WRITING IS DESIRED. THE CIRCUITS ARE WRITTEN           HC  470
C          AS ORDERED SEQUENCES OF  N  VERTICES.                        HC  480
C                                                                       HC  490
C MEANING OF THE INPUT-OUTPUT PARAMETERS:                               HC  500
C NC(INPUT)  = UPPER BOUND ON THE NUMBER OF HAMILTONIAN                 HC  510
C              CIRCUITS TO BE FOUND ( NC = - 1 IF ALL THE               HC  520
C              HAMILTONIAN CIRCUITS ARE TO BE FOUND).                   HC  530
C NC(OUTPUT) = NUMBER OF HAMILTONIAN CIRCUITS FOUND.                    HC  540
C NB(INPUT)  = - 1  IF HC MUST BE EXECUTED AS AN EXACT                  HC  550
C              PROCEDURE.                                               HC  560
C            = UPPER BOUND ON THE NUMBER OF BACKTRACKINGS IF            HC  570
C              HC MUST BE EXECUTED AS A HEURISTIC PROCEDURE.            HC  580
C NB(OUTPUT) = NUMBER OF BACKTRACKINGS PERFORMED. WHEN HC               HC  590
C              HAS BEEN EXECUTED AS A HEURISTIC PROCEDURE,              HC  600
C              IF  NB(OUTPUT) .LT. NB(INPUT)  THEN THE                  HC  610
C              RESULT OBTAINED IS EXACT.                                HC  620
C                                                                       HC  630
C MEANING OF THE OUTPUT PARAMETER:                                      HC  640
C S(I) = I-TH  VERTEX IN THE LAST HAMILTONIAN CIRCUIT FOUND.            HC  650
C                                                                       HC  660
```

```
C ON RETURN OF HC  N, PR  AND  KW  ARE UNCHANGED, WHILE IN                HC   670
C AR  THE ORDER OF THE ELEMENTS WITHIN EACH RECORD MAY BE                 HC   680
C ALTERED.                                                                HC   690
C                                                                         HC   700
C MEANING OF THE WORK ARRAYS:                                             HC   710
C PC(I)    = SUM OF THE IN-DEGREES OF VERTICES  1, ..., I-1               HC   720
C            ( PC(1) = 0 ).                                               HC   730
C AC       = ADJACENCY LIST (BACKWARD). THE ELEMENTS FROM                 HC   740
C            AC(PC(I)+1)  TO  AC(PC(I+1))  CONTAIN, IN ANY                HC   750
C            ORDER, ALL THE VERTICES  J  SUCH THAT ARC  (J,I)             HC   760
C            EXISTS.                                                      HC   770
C WHEN AN ARC IS REMOVED FROM THE GRAPH AT THE  K-TH  LEVEL               HC   780
C OF THE BRANCH-DECISION TREE, THE CORRESPONDING ELEMENTS                 HC   790
C AR(Q)  AND  AC(T)  ARE SET TO  - (K*(N+1) + AR(Q))  AND                 HC   800
C TO  - (K*(N+1) + AC(T)) , RESPECTIVELY.                                 HC   810
C VR(I)    = CURRENT OUT-DEGREE OF VERTEX  I .                            HC   820
C VC(I)    = CURRENT IN-DEGREE OF VERTEX  I .                             HC   830
C SUBR(I)  = - (K*(N+1) + J)  IF ARC  (I,J)  WAS IMPLIED AT               HC   840
C            THE  K-TH  LEVEL OF THE BRANCH-DECISION TREE.                HC   850
C          = 0  OTHERWISE.                                                HC   860
C RBUS(I)  = - J  IF ARC  (J,I)  IS CURRENTLY IMPLIED.                    HC   870
C          = 0  OTHERWISE.                                                HC   880
C TOR(K)   = Q*(M+1) + T  IF THE ARC GOING FROM  S(K)  TO THE             HC   890
C            ROOT, CORRESPONDING TO  AR(Q)  AND TO  AC(T),                HC   900
C            WAS REMOVED FROM THE GRAPH AT THE  K-TH  LEVEL               HC   910
C            OF THE BRANCH-DECISION TREE.                                 HC   920
C          = 0  OTHERWISE.                                                HC   930
C P(I)     = POINTER FOR THE FORWARD STEP. THE NEXT ARC                   HC   940
C            STARTING FROM  I  TO BE CONSIDERED IN THE                    HC   950
C            BRANCH-DECISION TREE IS  (I,AR(PR(I)+P(I)).                  HC   960
C                                                                         HC   970
C MEANING OF THE MAIN WORK SIMPLE VARIABLES:                              HC   980
C JR  = ROOT. THE HAMILTONIAN CIRCUITS ARE DETERMINED AS                  HC   990
C       PATHS STARTING AND ENDING AT  JR .                                HC  1000
C K   = CURRENT LEVEL OF THE BRANCH-DECISION TREE.                        HC  1010
C M   = NUMBER OF ARCS.                                                   HC  1020
C MP1 = M + 1 (USED FOR PACKING  TOR ).                                   HC  1030
C NP1 = N + 1 (USED FOR PACKING  AR ,  AC  AND  SUBR )                    HC  1040
C                                                                         HC  1050
      INTEGER PR(NP1), PC(NP1), AR(M), AC(M), S(N), VR(N), VC(N), P(N),   HC  1060
     * SUBR(N), RBUS(N), TOR(N)                                           HC  1070
```