# A Search Procedure for Hamilton Paths and Circuits

FRANK RUBIN

*IBM Corporation, Poughkeepsie, New York*

ABSTRACT. A search procedure is given which will determine whether Hamilton paths or circuits exist in a given graph, and will find one or all of them. A combined procedure is given for both directed and undirected graphs. The search consists of creating partial paths and making deductions which determine whether each partial path is a section of any Hamilton path whatever, and which direct the extension of the partial paths.

KEY WORDS AND PHRASES: Hamilton path, Hamilton circuit, heuristic search, longest path

CR CATEGORIES: 3.66, 5.32

## 1. *Introduction*

Techniques for finding Hamilton circuits and paths are fairly numerous. Most involve exhaustive searches, carried out sequentially or in parallel, which eliminate partial paths only when they double back upon themselves [1, 3–10].

Hakimi [2] improves upon these methods by the addition of deduction rules which allow earlier termination of partial paths, and elimination of certain edges from consideration. (These are rules R2, D1, F1, and F2 of Section 5 of this paper.) The present paper extends this method by adding some additional deduction rules (D3, F5, F6, F7, F8, and F9) and extending the method to directed as well as undirected graphs.

## 2. *Terminology*

Generally, standard graph theory terminology has been used throughout. The term graph is understood to mean a finite directed loopfree monovalent connected graph, an ordered pair $G = (V, E)$ where $V$ is a finite set of elements called vertices or nodes, and $E$ is a set of ordered pairs of elements in $V$ called arcs or edges. Whenever two directed arcs $(a, b)$ and $(b, a)$ are present, the pair will be regarded as a single undirected arc.

## 3. *Equivalence of Problems*

The remainder of this paper will deal with finding Hamilton circuits. The problem of finding Hamilton paths is made equivalent by adding a new node connected to every other node by an undirected arc. In an equally simple way, the constrained Hamilton path problems, where one or both endpoints of the path are given, can be reduced to finding a Hamilton circuit.



Author's address: IBM Corporation, Department B62—Building 707, Box 390, Poughkeepsie, NY 12602.

## 4. *The Search Procedure*

The basic search procedure is as follows:

S1. Select any single node as the initial path.
S2. Test the path for admissibility.
S3. If the path so far is admissible, list the successors of the last node chosen, and extend the path to the first of these. Repeat step S2.
S4. If the path so far is inadmissible, delete the last node chosen and choose the next listed successor of the preceding node. Repeat step S2.
S5. If all extensions from a given node have been shown inadmissible, repeat step S4.
S6. If all extensions from the initial node have been shown inadmissible, then no circuit exists.
S7. If a successor of the last node is the origin, a Hamilton circuit is formed; if all Hamilton circuits are required, then list the circuit found, mark the partial path inadmissible, and repeat step S4.

The test for admissibility in step S2 is made by the deduction rules in Section 5.

## 5. *Deductions*

Testing a partial path, $P$, for admissibility consists of classifying the edges of the graph into three sets: $D$ = deleted edges—those edges which cannot be in any Hamilton path containing the partial path; $R$ = required edges—those edges which must be in every Hamilton path containing the partial path; $U$ = undecided edges—those edges which cannot yet be classified. The edges in $D$, $R$, and $U$ are mutually exclusive, $D \cup R \cup U = E$, and the edges in $P$ are also in $R$.

Classification of the edges is performed by applying the following rules. The rules may be applied in any order, and are applied at each stage until exhaustion.

R. *Required edge rules:*

R1. If a vertex has only one directed arc entering (leaving), then that arc is required.

R2. If a vertex has only two arcs incident, then both arcs are required.

A. *Direction assignment rules:*

A1. If a vertex has a required directed arc entering (leaving), then all incident undirected arcs are assigned the direction leaving (entering) that vertex.

A2. If a vertex has a required undirected arc incident, and all other incident arcs are leaving (entering) the vertex, then the required arc is assigned the direction entering (leaving) the vertex.

D. *Deleted edge rules:*

D1. If a vertex has two required arcs incident, then all undecided arcs incident may be deleted.

D2. If a vertex has a required directed arc entering (leaving), then all undecided directed arcs entering (leaving) may be deleted.

D3. Delete any arc which forms a closed circuit with required arcs, unless it completes the Hamilton circuit.

F. *Failure, or termination, rules:*

F1. Fail if any vertex becomes isolated, that is, has no incident arc.

F2. Fail if any vertex has only one incident arc.

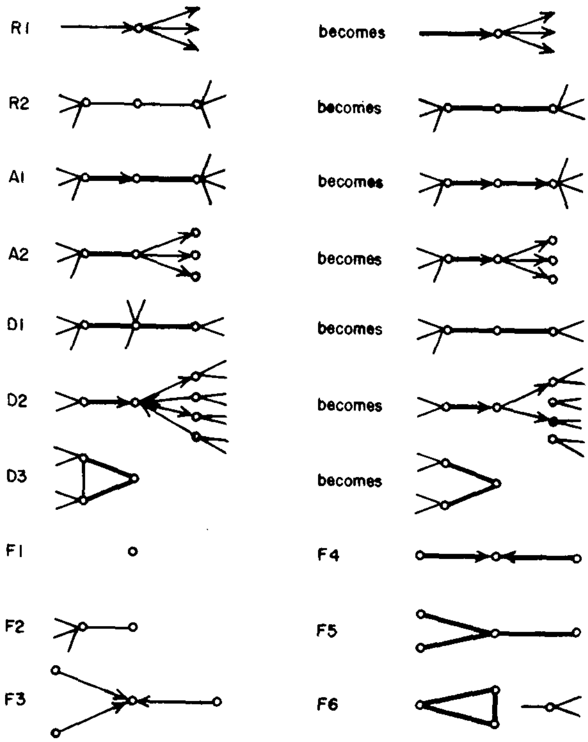
F3. Fail if any vertex has no directed arc entering (leaving).

F4. Fail if any vertex has two required directed arcs entering (leaving).

F5. Fail if any vertex has three required arcs incident.

F6. Fail if any set of required arcs forms a closed circuit, other than a Hamilton circuit.

Whenever a failure rule applies, the partial path is inadmissible and step S4 of the search procedure is taken. The deduction rules above are illustrated in Figure 1.




Undecided arc, set $U$.

Required arc, set $R$.

FIG. 1. Deduction rules

Since the algorithm and deduction rules are oriented toward directed graphs, the Hamilton circuits in an undirected graph will be generated twice each, with the nodes named in opposite order. To prevent this redundancy, in step S3 of the search the successors of the origin node may be numbered. Then the undirected arcs to successor nodes which are numbered lower than the successor presently being considered should be deleted. Thus if 0 is the origin and $1, 2, \cdots, K$ are its successors, delete arcs $(0, 1)$, $(0, 2), \cdots, (0, i-1)$ when considering successor $i$.

## 6. *Connectivity*

All of the preceding rules are quadratic in computational nature, in that they involve examining only the arcs incident at one node, or a list of nodes without repetitions, with each operation repeated for each such node. A test for the simple connectivity of the graph will involve examining each arc in the connected component containing the partial path, hence may also be quadratic in the number of nodes. Since the term "connected" is used in several senses for directed graphs, two failure rules result:

F7. *Fail if for any node not in the partial path there is no directed path to that node from the last node in the partial path.*

F8. Fail if for any node not in the partial path there is no directed path from that node to the initial node of the partial path.

A simple algorithm for rule F7 involves a list of nodes reachable by directed paths, and a set of flags for nodes which have already been reached. The rule F8 may be tested by the same algorithm, reversing all arc directions including those along the partial path.

C1. Flag all the nodes in the partial path.

C2. Place the last node on the partial path in the list.

C3. Choose any node on the list. Delete it from the list. Place all of its unflagged successors on the list, and flag them.

C4. Repeat step C3 until the list is empty. If every node is flagged, then the partial path is admissible; otherwise, it is inadmissible.

## 7. *Decomposition and Reduction*

Sometimes a graph has special connectivity properties which allow it to be reduced before presentation to the computer. The $k$-connectivity argument is applied to the *underlying graph* formed by replacing each directed arc $(a, b) \in R \cup U$ with an undirected arc.

A graph is called *$k$-connected* if the removal of some set of $k$ nodes and their incident arcs leaves the graph disconnected, but no set of $k - 1$ nodes disconnects the graph. The disconnecting nodes are called *$k$-articulation* nodes. The connected components of the graph formed by deleting the $k$-articulation nodes are called *interior $k$-components.* The subgraph formed by deleting an interior $k$-component and its incident edges is called an *exterior $k$-component.*

When $k = 1$, there is a single disconnecting node. Once the partial path includes this node, rule F8 will apply to the terminal node of the partial path. Consequently:

F9. Fail if the graph is 1-connected.

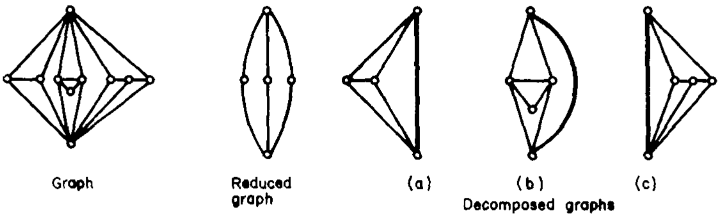



FIG. 2. Decomposition and reduction

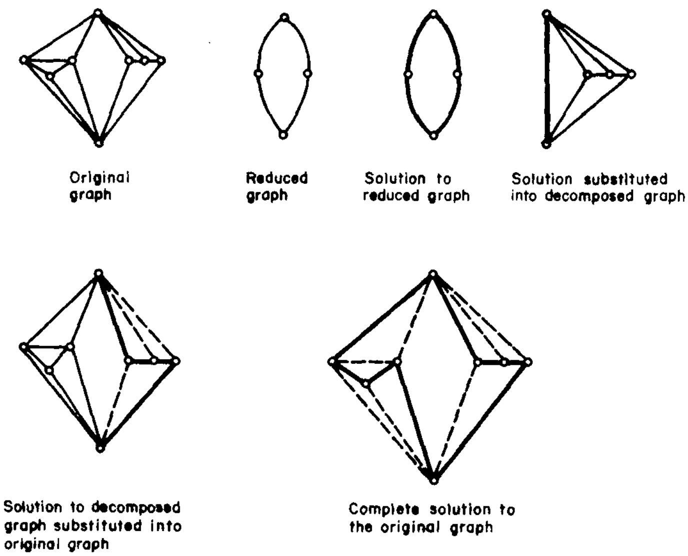



FIG. 3. Composition of circuits

When $k > 1$ the graph may be reduced by replacing each interior $k$-component by a canonical reduced form. The reduced graph is solved; then each component is solved separately. The solutions then are composed to solve the complete graph. For $k = 2$, the canonical form is a single node with two adjacent arcs. For $k > 2$ the canonical form is the complete $k$-graph. However, for $k = 3$, if at each 3-articulation node there is just one external arc, or all of the external arcs are entering (leaving), then the canonical form is a single node.

The solution by decomposition and reduction is illustrated in Figure 2. Composition of these solutions is shown in Figure 3.

REFERENCES

1. Danielson, Gordon H. On finding simple paths and circuits in a graph. *IEEE Trans. Circuit Theor. 15* (1968), 294–295.
2. Hakimi, S. L. Recent progress and new problems in applied graph theory. IEEE Region Six Conference Record, 1966, pp. 635–643.
3. Kamae, Takahiko. A systematic method of finding all directed circuits and enumerating all directed paths. *IEEE Trans. Circuit Theor. 14* (1967), 166–171.
4. Kroft, D. All paths through a maze. *Proc. IEEE 55* (1967), 88–90.
5. Parthasarathy, K. R. Enumeration of all paths in digraphs. *Psychometrika 29* (1964), 152–165.
6. Ponstein, J. Self-avoiding paths and the adjacency matrix of a graph. *J. SIAM 14* (1966), 600–609.
7. Rao, V. V. Bapeswara, and Murti, V. G. K. Enumeration of all circuits in a graph. *Proc. IEEE 57* (1969), 700–701.
8. Rao, V. V. Bapeswara, and Murti, V. G. K. Enumeration of all circuits or cutsets of a graph. *Electronics Letters 6* (Feb. 19, 1970), 104–105.
9. Roberts, S. M., and Flores, Benito. Systematic generation of Hamiltonian circuits. *Comm. ACM 9*, 9 (Sept. 1966), 690–694.
10. Yau, S. S. Generation of all Hamiltonian circuits, paths, and centers of a graph, and related problems. *IEEE Trans. Circuit Theor. 14* (1967), 79–81.

RECEIVED APRIL 1973; REVISED OCTOBER 1973