A Polynomial-Time Algorithm for Detecting Hamiltonian Cycles in Undirected Graphs

Author: [Liang] [Chen]

1 Abstract

The algorithm in this paper is primarily designed to identify a Hamiltonian cycle in graphs confirmed to contain one.

The core idea involves removing redundant edges while preserving essential ones; the remaining edges after deletion form the Hamiltonian cycle.

The algorithm's greatest advantage is that it requires no backtracking.

2 Keywords

2.1 Marked in Red:

In the algorithm, edges confirmed to be preserved are marked in red. Red-marked edges are part of a Hamiltonian cycle and thus are not deleted. Red-marked vertices have only two edges. Red-marked edges and vertices need not be traversed again.

2.2 Red-Marked Edges of a Vertex:

The number of red-marked edges among all edges connected to a given vertex.

2.3 Red-Marked Adjacent Edges of an Edge:

Every edge has two adjacent vertices. All edges connected to these vertices (excluding the edge itself) are it's adjacent edges; those marked in red are it's red-marked adjacent edges.

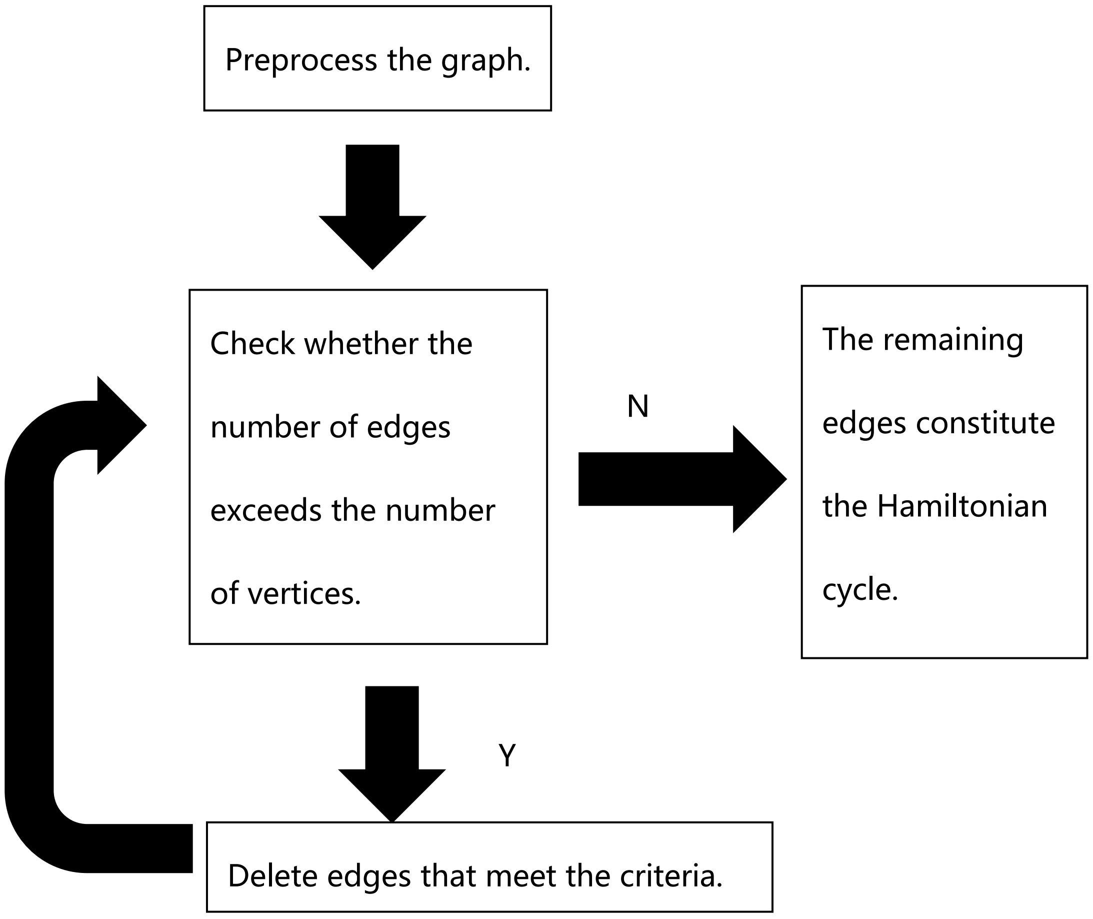
2.4 Self-Loop:

In directed or undirected graphs, an edge that originates from and returns to the same vertex.

2.5 Parallel Edges:

In undirected graphs, two edges share the same pair of vertices.

3 Algorithm Description


Preprocess the graph.
Check whether the number of edges exceeds the number of vertices.
N
The remaining edges constitute the Hamiltonian cycle.
Y
Delete edges that meet the criteria.

3.1 Graph Preprocessing

Traverse all vertices and edges, eliminating all parallel edges and self-loops. Self-loops are deleted directly; for parallel edges, only one is retained, and the rest are deleted.

After deletion, traverse again. If any vertex has a degree of 2, mark its two edges and the vertex itself in red.

3.2 Check if Edge Count Exceeds Vertex Count.

After deleting unnecessary edges, the resulting Hamiltonian cycle must satisfy: the number of edges equals the number of vertices.

3.3 Delete Edges Meeting Criteria

In both directed and undirected graphs, after any deletion, halt further deletions and proceed to Step 3.2. Deletions resume only if the condition is met.

Traverse all vertices. If any vertex has two red-marked edges, delete all its unmarked edges (multiple edges may be deleted at once). If a vertex has only two edges (some unmarked), mark all its edges and the vertex in red.

If no deletions occur, traverse all edges and delete the edge with the most red-marked adjacent edges (only one edge at a time). If an edge has two red-marked adjacent edges, delete it immediately and proceed to Step 3.2. Otherwise, delete an edge with one red-marked adjacent edge, or any edge if none exist.

3.4 The Remaining Edges Form the Hamiltonian Cycle

4 Complexity Analysis

Let n be the number of vertices.

3.1.1:T(n)=$n + n^2$=O(n^2)

3.2:T(n)=O(1)

3.3.1:T(n)=($n^2 + n^4$) $\times$ ($n + n^2$)=O(n^6)

T(n)=$n^2 \times 1 \times n^6$=O(n^6)

5 Proof

In an undirected Hamiltonian graph, if a vertex has only two edges, every Hamiltonian cycle must include both edges. Thus, such vertices and edges can be treated as a single undeletable unit. For multiple vertices/edges, this holds if all Hamiltonian cycles pass through them uniquely.

Per "Studies on Minimally n-Connected Graphs" (R. Halin), deleting qualifying edges preserves Hamiltonicity, ensuring Step 3.3.1's deletions are valid.

6 References

1. Halin, R. "Studies on Minimally n-Connected Graphs."

7 Appendix (Original Chinese Text)

一、摘要

此文章中的算法主要用于在确定有哈密顿环的图中找出一个哈密顿环。

主要思想是删除多余的边，保留一定需要的边，将所有不需要的边删除后剩余的边组成哈密顿环。

此算法最大的优势是不需要回溯。

二、关键词

（一）标红：

在算法中会将确定不会删除的边标记为红色，已经标红的边是某哈密顿环的一部分所以不删除。已经标红的顶点只有两条边。已经标红的边和顶点不需要再次遍历。

（二）顶点的标红边：

某个顶点拥有的所有边中标红边的数量。

（三）边的标红临边：

任意一条边都有两个相邻顶点，这两个顶点拥有的除此边以外的所有边均为此边的临边，其中标红的边为此边的标红临边。

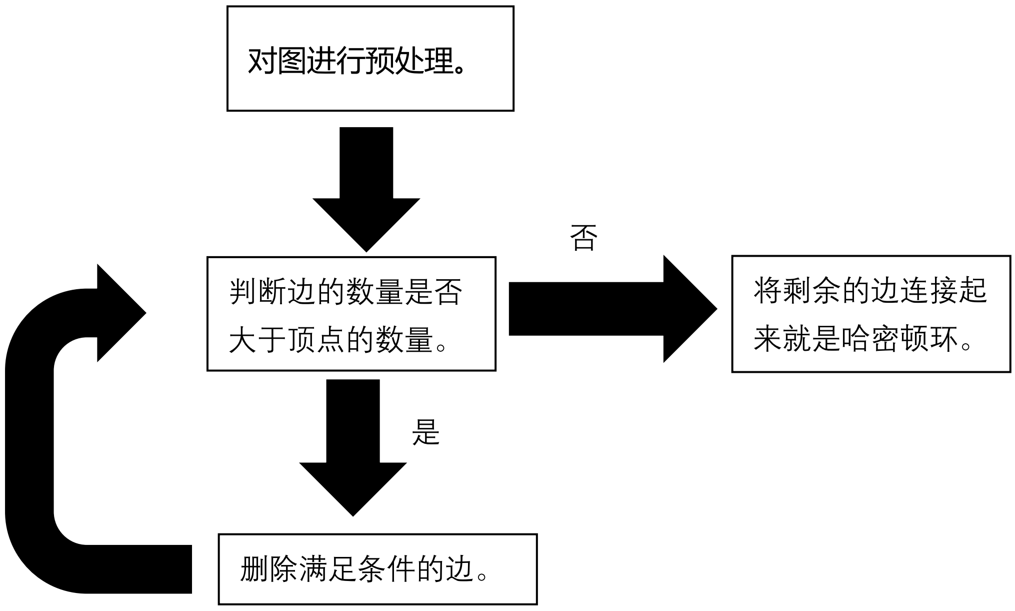
（四）自环：

在有向图和无向图中一条边从一个顶点出发又回到同一个顶点。

（五）平行边：

无向图中两条边所属的顶点相同。

三、算法描述


对图进行预处理。
否
判断边的数量是否大于顶点的数量。
将剩余的边连接起来就是哈密顿环。
是
删除满足条件的边。

（一）对图进行预处理

遍历此图中所有顶点和边，消除所有平行边和自环。自环的边直接删除，平行边只保留其中一条其余全部删除。

删除后再次遍历所有边和顶点，如果发现任意一个顶点的度数为 2，则将此顶点的两条边及其标红。

（二）判断边的数量是否大于顶点的数量

将所有不需要的边删除后形成的哈密顿环，顶点的数量和边的数量一定相同，剩余的边也就会组成一个哈密顿环。

（三）删除满足条件的边

在有向图和无向图中，一但发生一次删除行为，则需要停止任何删除行为，进行 3.2 步骤的判断。只有判断后满足条件才能再次删除。

遍历所有顶点，如果发现任意顶点的标红边数量为 2，

则将其余未标红的所有边删除，可以一次性删除多条边。如果发现任意一个顶点只有两条边且有未标红的边，则将其所有边及其顶点标红。

如果上一步没有任何删除行为，遍历所有边，则删除标红临边数量最多的任意一条边，不能一次性删除多条边。也就是发现任意一条边的标红临边数为 2 可立刻将其删除然后进行 3.2 步骤的判断，如果没有则删除任意一条标红临边数量为 1 的边，如果还没有则删除任意一条边。

（四）将剩余的边连接起来就是哈密顿环

四、复杂度分析

n 为顶点数量。

3.1.1:T(n)=$n+n^2$=O(n^2)

3.2:T(n)=O(1)

3.3.1:T(n)=$(n^2+n^4)\times(n+n^2)$=O(n^6)

T(n)=$n^2\times 1\times n^6$=O(n^6)

五、证明

在无向哈密顿图中，如果某个顶点只有两条边，那么无论哈密顿环怎么画都会经过这两条边，不存在其他可能性。所以在无向哈密顿图中如果某个顶点只有两条边，可以将这两条边和这个顶点视作一个整体，即可视作一条不可以删除的边。多个顶点和多条边的情况下，只要满足任意哈密顿环经过此点只有一种可能性，也可将其视为一条不可删除的边。

根据文章“Studies on minimally n-connected graphs”中的结论，删除满足条件的任意一条边后该图形依然是哈密顿的，所以 3.3.1 中的删边行为不会影响其哈密顿性。

六、参考文献

“Studies on minimally n-connected graphs”