Integer and Combinatorial Optimization: Hamiltonian Cycle is NP-Complete

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

January 2019
Hamiltonian cycles

Definition

A Hamiltonian cycle on a graph $G = (V, E)$ is a cycle that visits every vertex.

$1 - 2 - 3 - 5 - 8 - 6 - 7 - 4 - 1$

is a Hamiltonian cycle.
A graph without a Hamiltonian cycle

This graph does not have a Hamiltonian cycle
A graph without a Hamiltonian cycle

This graph does not have a Hamiltonian cycle

Look at vertices of degree 2
Hamiltonian cycle problem

Definition

Let $G = (V, E)$ be a graph. The Hamiltonian cycle feasibility problem is to determine whether there is a Hamiltonian cycle in $G = (V, E)$.

Theorem

The Hamiltonian cycle problem is $\mathcal{NP}$-complete.

First show the problem is in $\mathcal{NP}$:

Our certificate of feasibility consists of a list of the edges in the Hamiltonian cycle.

We can check quickly that this is a cycle that visits every vertex.
Hamiltonian cycle problem

Definition

Let $G = (V, E)$ be a graph. The Hamiltonian cycle feasibility problem is to determine whether there is a Hamiltonian cycle in $G = (V, E)$.

Theorem

The Hamiltonian cycle problem is $NP$-complete.

First show the problem is in $NP$:

Our certificate of feasibility consists of a list of the edges in the Hamiltonian cycle.

We can check quickly that this is a cycle that visits every vertex.
Polynomially transform 3-SAT to Hamiltonian cycle

We show that 3-SAT can be polynomially transformed to the Hamiltonian cycle problem.

This requires the construction of two sets of gadgets, and specification of how to link them together.

We have an instance of 3-SAT with $n$ boolean variables $z_1, \ldots, z_n$ and $m$ clauses $C_1, \ldots, C_m$, with each clause containing exactly 3 literals.

\[ z_1 + z_4 + \overline{z}_8 \]

\[ \overline{z}_3 + z_5 + z_9 \]
Gadget to enforce clause satisfaction

The first type of gadget is to force satisfaction of a clause. Each clause is mapped into a gadget with 13 vertices.

There are multiple ways to traverse the gadget, each corresponding to different ways to satisfy the clause.

Any Hamiltonian cycle that enters at vertex $a$ and leaves at vertex $d$ is required to visit all the other vertices of the gadget in between.

It is not possible to traverse all three of the edges $(a, b), (b, c), (c, d)$.
Omit edge \((a, b)\)
Omit edge \((b, c)\)
Omit edges \((a, b)\) and \((b, c)\)
Omit edges \((a, b)\) and \((c, d)\)
Omit edges \((a, b), (b, c), \text{ and } (c, d)\)
Relating the gadget to a clause

Each of the edges \((a, b), (b, c), (c, d)\) corresponds to a literal in the clause.

**Not traversing** the edge corresponds to the literal taking the value **TRUE**.

It is **not possible to traverse all three of the edges** \((a, b), (b, c), (c, d)\), but any subset of the edges can be traversed.

Hence, any Hamiltonian cycle will correspond to each clause being satisfied.

Note also that if any of the edges \((a, b), (b, c), (c, d)\) is traversed, then it must be traversed from bottom to top.
Gadget B

Compact representation:
Putting the gadgets together

The gadgets $B$ are placed in series, one for each clause. A pair of edges is constructed for each variable, as shown in Figure 15-10 taken from the text by Papadimitriou and Steiglitz.

Traversing the left hand edge of the two corresponds to setting the boolean literal to TRUE, and traversing the right corresponds to setting it to FALSE. As long as at least one of the Gadget B right hand edges is not traversed, the clause is satisfied.
Edges for the variables

The Hamilton circuit shown corresponds to:

\[ F = (x_1 + \overline{x}_2 + x_3)(\overline{x}_1 + x_2 + \overline{x}_3)(\overline{x}_1 + \overline{x}_2 + x_3) \]

\[ t(x_1) = \text{true} \]
\[ t(x_2) = \text{false} \]
\[ t(x_3) = \text{false} \]
**Consistency** can be ensured by constructing another gadget.

If we traverse an edge corresponding to a literal taking the value FALSE, we need to ensure that the literal takes the value FALSE in every other clause where it appears.
Traversing gadget A

We can traverse the gadget in two ways:
(i) enter below $n$ and leave above $q$, or
(ii) enter above $y$ and leave below $v$.
Linking the gadgets

A copy of this gadget will be used for every literal.

The vertices \( n \) through \( q \) are placed on the right hand edges of the \( B \) gadgets.

If this right hand edge of gadget \( B \) corresponds to the unnegated literal \( z_j \) then the vertices \( y \) through \( v \) are placed on the TRUE side for variable \( z_j \) in the pair of edges on the right of the graph.

Conversely, if the right hand edge of the \( B \) gadget corresponds to the literal \( \overline{z}_j \) then the vertices \( y \) through \( v \) are placed on the FALSE side for variable \( z_j \) in the pair of edges on the right of the graph.
The Papadimitriou and Steiglitz example

\[ F = (x_1 + \overline{x}_2 + x_3) (\overline{x}_1 + x_2 + \overline{x}_3) (\overline{x}_1 + \overline{x}_2 + x_3) \]

The Hamilton circuit shown corresponds to:
\[
\begin{align*}
  t(x_1) &= \text{true} \\
  t(x_2) &= \text{false} \\
  t(x_3) &= \text{false}
\end{align*}
\]
The traveling salesman problem

Definition

An instance of the traveling salesman problem (TSP) with upper bound is defined by a complete graph $K_n$ with edge weights $w_e$ and an upper bound $B$. The answer to the feasibility problem is YES if there exists a Hamiltonian tour in the graph of length no greater than $B$; otherwise, the answer is NO.

Theorem

TSP with upper bound is NP-complete.

Any tour is a certificate of feasibility, so the problem is in $NP$. 
The traveling salesman problem

Definition

An instance of the traveling salesman problem (TSP) with upper bound is defined by a complete graph $K_n$ with edge weights $w_e$ and an upper bound $B$. The answer to the feasibility problem is YES if there exists a Hamiltonian tour in the graph of length no greater than $B$; otherwise, the answer is NO.

Theorem

TSP with upper bound is NP-complete.

Any tour is a certificate of feasibility, so the problem is in $NP$. 

Mitchell
Polynomially transform HC to TSP

We reduce Hamiltonian cycle to TSP with upper bound.

Given an instance of Hamiltonian cycle on graph $G = (V, E)$, we construct a TSP instance by using the same vertices and giving the edges of the complete graph the following weights:

$$w_e = \begin{cases} 
1 & \text{if } e \in E \\
2 & \text{if } e \notin E 
\end{cases}$$

The upper bound is chosen as $B = |V|$.

There exists a Hamiltonian cycle in the original graph if and only if there is a TSP tour which only uses edges with $w_e = 1$, so if and only if there is a TSP tour of length no greater than $|V|$.
Reducing HC to TSP

\[ w_e = 1 \]
\[ w_e = 2 \]
\[ B = 5 \]
Reducing HC to TSP

Mitchell

Hamiltonian Cycle is NP-Completes
The triangle inequality

Note that the edge lengths in the constructed TSP instance satisfy the triangle inequality:

\[ w_{ab} + w_{bc} \geq w_{ac} \]

for any vertices \( a, b, c \).

Hence, we’ve proved:

the restricted version of the TSP where the edge lengths must satisfy the triangle inequality

is also \( \mathcal{NP} \)-complete.
Hamiltonian cycle

Definition
A Hamiltonian path on a graph $G = (V, E)$ is a path that visits every vertex.

Definition
Let $G = (V, E)$ be a graph. The Hamiltonian path feasibility problem is to determine whether there is a Hamiltonian path in $G = (V, E)$.

Theorem
The Hamiltonian path problem is $NP$-complete.

Any Hamiltonian path is a certificate of feasibility, so the problem is in $NP$. 
Hamiltonian cycle

**Definition**

A Hamiltonian path on a graph \( G = (V, E) \) is a path that visits every vertex.

**Definition**

Let \( G = (V, E) \) be a graph. The Hamiltonian path feasibility problem is to determine whether there is a Hamiltonian path in \( G = (V, E) \).

**Theorem**

The Hamiltonian path problem is \( \mathcal{NP} \)-complete.

Any Hamiltonian path is a certificate of feasibility, so the problem is in \( \mathcal{NP} \).
Hamiltonian cycle

**Definition**

A Hamiltonian path on a graph $G = (V, E)$ is a path that visits every vertex.

**Definition**

Let $G = (V, E)$ be a graph. The Hamiltonian path feasibility problem is to determine whether there is a Hamiltonian path in $G = (V, E)$.

**Theorem**

The Hamiltonian path problem is \(\mathcal{NP}\)-complete.

Any Hamiltonian path is a certificate of feasibility, so the problem is in \(\mathcal{NP}\).
Reduce HC to HP

We reduce an instance of Hamiltonian cycle on a graph $G = (V, E)$ to Hamiltonian path in two different ways.

(There are many ways to do this.)
First reduction from HC to HP

Pick any vertex $v \in V$, split it into two vertices $v_1$ and $v_2$.

If $(v, u) \in E$ then edges $(v_1, u)$ and $(v_2, u)$ are included in the new graph.

Introduce two new nodes $s$ and $t$ and two new edges $(s, v_1)$ and $(v_2, t)$.

The nodes $s$ and $t$ are leaves in the new graph, so they must be the endpoints of any Hamiltonian path in the new graph.

This Hamiltonian path directly returns a Hamiltonian cycle in the original graph.
First reduction from HC to HP: example

split vertex 3
First reduction from HC to HP: example

split vertex 3
First reduction from HC to HP: example

introduce leaves
First reduction from HC to HP: example

Hamiltonian cycle

Hamiltonian path

Mitchell

Hamiltonian Cycle is NP-Completes
Second reduction from HC to HP

The second reduction requires more work.

For each edge $e = (u, v)$ in $E$, perform the following steps:

1. Remove edge $e$ from $E$.
2. Add two new nodes $s$ and $t$ and two new edges $(s, u)$ and $(v, t)$.
3. Search for a Hamiltonian path in the modified graph. If one exists, it must have endpoints $s$ and $t$, so it must correspond to a Hamiltonian cycle in the original graph.

The number of calls to the Hamiltonian path algorithm is equal to the number of edges in the original graph with the second reduction.

Hence the $NP$-complete problem Hamiltonian cycle can be reduced to Hamiltonian path, so Hamiltonian path is itself $NP$-complete.
Second reduction from HC to HP

The second reduction requires more work.

For each edge \( e = (u, v) \) in \( E \), perform the following steps:

1. Remove edge \( e \) from \( E \).
2. Add two new nodes \( s \) and \( t \) and two new edges \( (s, u) \) and \( (v, t) \).
3. Search for a Hamiltonian path in the modified graph. If one exists, it must have endpoints \( s \) and \( t \), so it must correspond to a Hamiltonian cycle in the original graph.

The number of calls to the Hamiltonian path algorithm is equal to the number of edges in the original graph with the second reduction.

Hence the \( \mathcal{NP} \)-complete problem Hamiltonian cycle can be reduced to Hamiltonian path, so Hamiltonian path is itself \( \mathcal{NP} \)-complete.
Second reduction from HC to HP

The second reduction requires more work.

For each edge \( e = (u, v) \) in \( E \), perform the following steps:

1. Remove edge \( e \) from \( E \).
2. Add two new nodes \( s \) and \( t \) and two new edges \( (s, u) \) and \( (v, t) \).
3. Search for a Hamiltonian path in the modified graph. If one exists, it must have endpoints \( s \) and \( t \), so it must correspond to a Hamiltonian cycle in the original graph.

The number of calls to the Hamiltonian path algorithm is equal to the number of edges in the original graph with the second reduction.

Hence the \( NP \)-complete problem Hamiltonian cycle can be reduced to Hamiltonian path, so Hamiltonian path is itself \( NP \)-complete.
Second reduction from HC to HP: example

Remove edge (2, 4)
Second reduction from HC to HP: example

remove edge (2, 4)
Second reduction from HC to HP: example

introduce leaves
Second reduction from HC to HP: example

Hamiltonian cycle

no Hamiltonian path!
Second reduction: remove edge \((2, 3)\)
Second reduction: remove edge (2,3)
Second reduction: remove edge (2,3)

introduce leaves
Second reduction: remove edge \((2, 3)\)

Hamiltonian cycle

Hamiltonian path