# Hamiltonian Path is NP-Complete

CSC 463

March 5, 2020

## 1 Hamiltonian Path

A graph $G$ has a **Hamiltonian path** from $s$ to $t$ if there is an $s$ to $t$ path that visits all of the vertices exactly once. Similarly, a graph $G$ has a **Hamiltonian cycle** if $G$ has a cycle that uses all of its vertices exactly once. We will prove that the problem **D-HAM-PATH** of determining if a directed graph has an Hamiltonian path from $s$ to $t$ is **NP**-Complete.

**Theorem 1.** **D-HAM-PATH** *is* **NP**-*Complete.*

*Proof.* Let $G$ be a directed graph. We can check if a potential $s, t$ path is Hamiltonian in $G$ in polynomial time. Now, we will give a polynomial-time reduction **3SAT** $\leq_p$ **D-HAM-PATH** to complete the proof of completeness.

Suppose a conjunctive normal form formula $\phi = \bigwedge_{i=1}^{m} \phi_i$ has $n$ variables and $m$ clauses. To simplify the proof, we may assume that no clause in $\phi$ contains both a variable $x_i$ and its negation $\overline{x_i}$, since any clause containing both literals can be removed with $\phi$ without affecting its satisfiability. The reduction uses the following steps.

1. First, for every variable $x_i$, we generate $2m+1$ vertices labelled $v_{i,j}$ and add directed edges $(v_{i,j}, v_{i,j+1})$ and $(v_{i,j+1}, v_{i,j})$ for $0 \leq j \leq 2m$.

2. Next, we connect vertices associated to different variables by adding four directed edges $(v_{i,0}, v_{i+1,0})$, $(v_{i,0}, v_{i+1,2m+1})$, $(v_{i,2m+1}, v_{i+1,0})$, $(v_{i,2m+1}, v_{i+1,2m+1})$.

3. Next, we create a vertex labelled $c_j$. If $x_i$ appears in clause $\phi_j$ positively, add directed edges $(v_{i,2j-1}, c_j)$ and $(c_j, v_{i,2j})$. Otherwise, if $x_i$ appears negatively in clause $\phi_j$, add directed edges $(v_{i,2j}, c_j)$ and $(c_j, v_{i,2j-1})$.

4. Finally, we add two extra vertices $s$ and $t$ connect $s$ by adding $(s, v_{1,1}), (s, v_{1,2m+1})$ and connect $t$ by adding $(v_{n,1}, t)$ and $(v_{n,2m+1}, t)$. Call the graph generated $G_\phi$. See Figure 1 for what the resulting graph looks like.

Now, we have to show that $\phi$ is satisfiable if and only if there is Hamiltonian path in $G_\phi$ from $s$ to $t$. Suppose $\phi$ is satisfiable. Then we can visit each variable vertex starting from $s$ by going from $v_{i,0}$ to $v_{i,2m+1}$ from left to right if $x_i$ is assigned true, and going from $v_{i,2m+1}$ to $v_{i,0}$ from right to left if $x_i$ is assigned false, and ending at $t$ after $v_{n,0}$ or $v_{n,2m+1}$ is visited. Furthermore, each clause vertex can be visited since by assumption, each clause $\phi_i$ has some literal $l_i = x_k$ or $l_i = \overline{x_k}$ that is assigned true. Each vertex $c_j$ by can visited by using the edges $(x_{k,2j-1}, c_j)$ and $(c_j, x_{k,2j})$ in the first case, and $(x_{k,2j}, c_j)$ and $(c_j, x_{k,2j-1})$ in the second case, since the path goes right in the graph for variables assigned true and the path goes left for variables assigned false. Thus, $G_\phi$ has a Hamiltonian path if $\phi$ is satisfiable.

Conversely, suppose $G_\phi$ has an $(s,t)$ Hamiltonian path $P$. Notice that $G_\phi$ without the clause vertices is Hamiltonian so we need to show that a path doesn't "get stuck" after visiting a clause vertex. More formally, it suffices to show that if $P$ visits $v_{i,2j-1}, c_j, v$ in that order, then $v = v_{i,2j}$. Suppose $v \neq v_{i,2j}$. Then notice that the only vertices coming into $v_{i,2j}$ are $v_{i,2j-1}, c_j, v_{i,2j+1}$ and the only vertices coming out of $v_{i,2j}$ are to $v_{i,2j-1}$ and $v_{i,2j+1}$. Hence $v_{i,2j}$ must be visited coming in from $v_{i,2j+1}$, but now the path is stuck and cannot continue since $v_{i,2j-1}$ has already been visited. A similar argument shows that if $P$ visits $v_{i,2j}, c_j, v$ in that

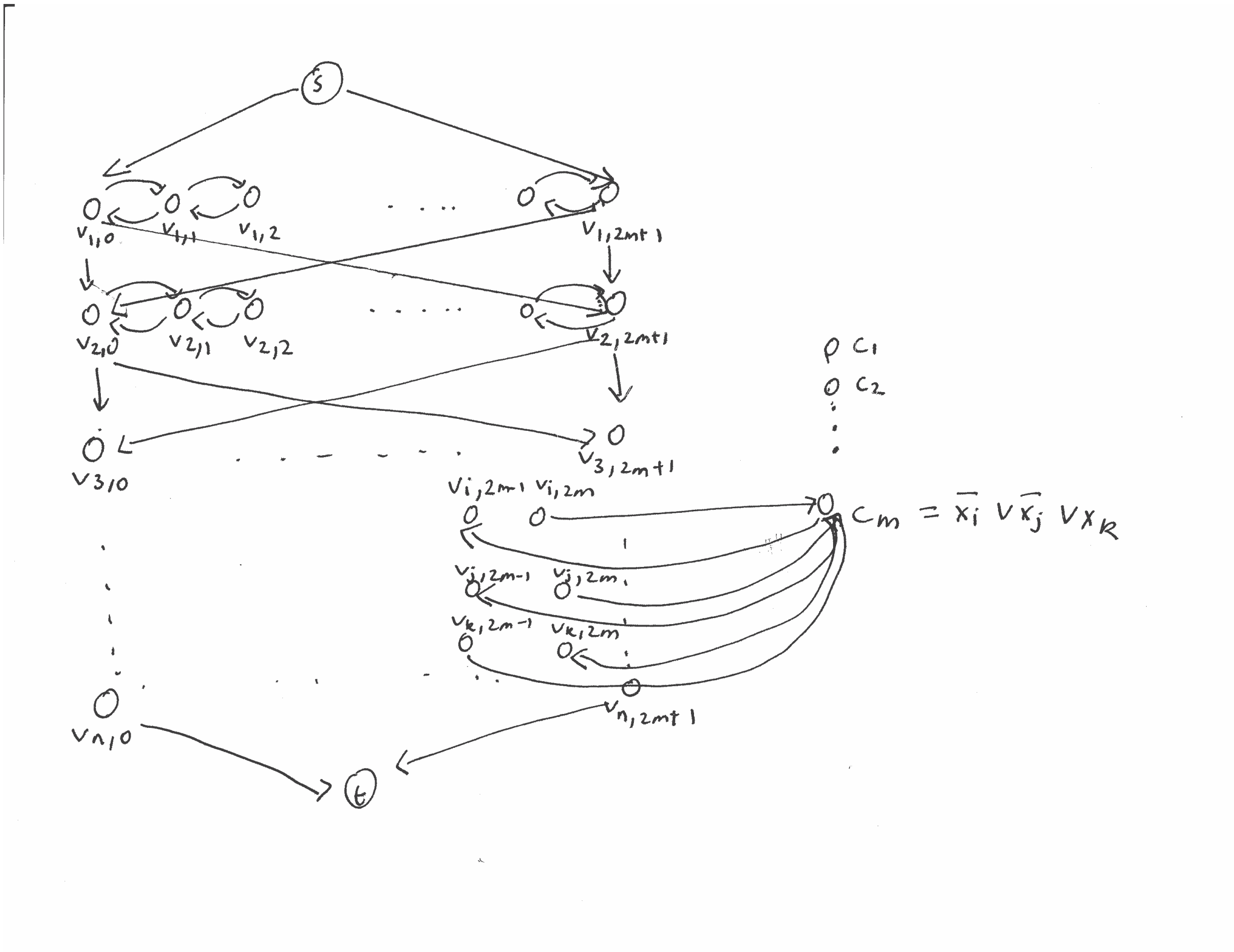

Figure 1: Reduction from 3SAT to HAM-PATH

order, then $v = v_{i,2j-1}$. So a Hamiltonian path in $G_\phi$ visits variable vertices in order from the vertices for $x_1$ to the vertices for $x_n$, interleaving the clause vertices in between variables associated to the same variable. Therefore, an assignment to the variables $x_i$ is well defined by noticing which direction the path takes along the variable vertices for $x_i$ in $G_\phi$. By construction, this assignment will satisfy $\phi$ since the assignment makes a literal in each clause true.

The reduction runs in $O(mn)$ time which is a polynomial in the length of the input. □

## 2 Hardness of Approximation for TSP

Once we have proved that the directed Hamiltonian path problem is **NP**-Complete, then we can use further reductions to prove that the following problems are also **NP**-Hard:

- Finding a Hamiltonian path in an **undirected** graph (by reduction from the directed problem).
- Finding a Hamiltonian cycle in an undirected graph (by reduction from Hamiltonian path).
- Finding a minimum weight Hamiltonian cycle in an weighted, complete graph (by reduction from the previous problem).

The last problem is also known as the **Travelling Salesperson Problem (TSP)**. It is given that name since it models the following situation. Given a list of cities and some cost of travelling between each pair of cities (eg. airfare, distance), then a minimum weight Hamiltonian cycle in this graph is a path visiting all of the cities exactly once and returns the origin city with least cost.

**TSP** is a problem that arises in many applications, so while we do not expect to find an efficient algorithm for it due of its **NP**-hardness, researchers have developed **approximation algorithms** to solve the problem in certain special cases. This is a general approach when facing **NP**-hard search problems. We cannot expect to find an optimal solution in polynomial time so all we can hope for is to find a good enough approximation in reasonable time. Unfortunately, there are **hardness of approximation** results that state that even searching for an approximation is impossible in polynomial time assuming that $\mathbf{P} \neq \mathbf{NP}$.

**Theorem 2.** *Assuming that* $\mathbf{P} \neq \mathbf{NP}$, *there is no polynomial time algorithm that when given a weighted graph finds a TSP tour that is at most 2 times the shortest tour.*

*Proof.* Let $G$ be an undirected graph with $n$ vertices. We generate a weighted complete graph $H$ based on $G$ with weights $w_{ij} = 1$ if edge $(i, j)$ exists in $G$, and weight $w_{ij} = n+2$ if no edge $(i, j)$ exists in $G$. Observe that the shortest TSP tour in $H$ has weight $n$ if $G$ has a Hamiltonian cycle, and otherwise, $H$ has weight at least $2n + 1$.

Hence, if there is an polynomial time algorithm $A$ that 2-approximates a TSP tour in $H$, we can decide if $G$ has a Hamiltonian cycle or not, since $G$ has a Hamiltonian cycle iff $A$ outputs a number less than or equal to $2n$. Therefore, if $A$ runs in polynomial-time, we can decide if an undirected graph has a Hamiltonian cycle in polynomial time, which is impossible assuming $\mathbf{P} \neq \mathbf{NP}$. □

Now assume that the weights in a TSP instance satisfy the triangle inequality $w_{ij} \leq w_{ik} + w_{kj}$ given vertices $i, j, k$. We call this new problem **Metric-TSP** when the weights have this constraint, and note that it is still **NP**-hard by reduction by undirected Hamiltonian cycle. However, with this restriction, we **can** now find a 2-approximation using the following algorithm.

**Algorithm 1.** Given a metric TSP instance $G$,

1. Firstly find a minimum weight spanning tree $T$ in $G$.
2. Then add a parallel edge to each edge $(u, v)$ in the tree $T$, and find an Euler cycle $E$ (a cycle visiting each edge exactly once) using these parallel edges.
3. Convert the Euler cycle $E$ in the previous step into a Hamiltonian cycle $H$ by skipping vertices that have already been visited. (Eg. if the Euler tour visits $v_1, v_2, v_1, v_3 \ldots$, the Hamiltonian cycle visits $v_1, v_2, v_3, \ldots$)

Note that each step runs in polynomial time and the algorithm is guaranteed to terminate, since we know that an Euler tour in a graph exists iff each vertex is adjacent to an even number of edges. Now we argue that this algorithm provides a 2-approximation.

**Theorem 3.** *Metric-TSP can be 2-approximated.*

*Proof.* Let $w(H), w(E), w(T)$ be the total weight of the Hamiltonian cycle, Euler tour, and minimum spanning tree produced by the algorithm, and let $O$ be the optimal TSP tour and $w(O)$ be its weight.

We know that $w(H) \leq w(E)$ since skipping edges always reduces the weight using the triangle inequality. Next, we have that $w(E) = 2w(T)$ since the Euler tour was created by doubling each edge in $T$. Finally, we have $w(T) \leq w(O)$ since deleting an edge from $O$ produces a spanning tree, and we know that $w(T)$ is the weight of the minimum spanning tree.

Hence, combining these inequalities together:

$$w(H) \leq w(E) = 2w(T) \leq 2w(O),$$

which means that we have produced a Hamiltonian cycle $H$ of weight at most 2 times the shortest cycle. □

In fact, it is known that **Metric-TSP** can be $\frac{3}{2}$-approximated, but improving this approximation only with the metric assumption has remained an open problem for around 45 years. Adding additional assumptions such as assuming that the metric is the usual Euclidean metric in $\mathbb{R}^d$, not just an arbitrary metric satisfying the triangle inequality, also improves the approximation factor.