

Proving Hamiltonian Circuit NP-Completeness

Ayob Asrout Vargas,
Dario Cerviño Luridiana,
Enrique Viña Alonso,
Juan García Santos

November 16, 2022

Contents

1	Introduction	2
1.1	Description of the problem	2
2	Proving the NP-completeness	2
2.1	Introduction	2
2.2	Vertex Cover Problem	2
2.3	Hamiltonian Circuit Problem	3
2.4	Proving Hamiltonian Circuit \in NP	3
2.5	Polynomial Transformation	3
2.5.1	Selectors	3
2.5.2	Cover-Testing components	3
2.5.3	Joining the components	4
2.5.4	Final demonstration	4
2.6	Sources	6

1 Introduction

1.1 Description of the problem

In this report, we will prove the NP-Completeness of the Hamiltonian Circuit problem, to do so, we will find a way to transform the input of another NP-Complete problem (in our case the Vertex Cover problem) into the input of a Hamiltonian Circuit Problem. The procedure used to transform the input must be polynomial and the resulting input for the Hamiltonian Circuit problem must be satisfied if and only if the original input satisfies the Vertex Cover problem.

2 Proving the NP-completeness

2.1 Introduction

We define the class of NP Languages as the languages that represent all of the encoded decision problems that a Non-Deterministic Turing Machine (NDTM) can solve in a polynomial time complexity.

The class of languages NP-Complete is defined by the following statement:

A language L is NP-Complete if it satisfies the following conditions:

$$L \in NP$$

$$\forall L' \in NP \quad \exists L' \propto L$$

This means, if the language L is NP-complete, it must belong to the NP class and every other language L' that belongs to the NP class must have a polynomial transformation to L .

Furthermore, if a language L' is known to be NP-complete, a polynomial transformation to L confirms L is also NP-complete. In formal terms:

$$L \in NP$$

$$L' \in NP_{Complete}$$

$$\exists L' \propto L \implies L \in NP_{Complete}$$

Knowing this, we can prove that the Hamiltonian Circuit problem is NP-Complete, by proving that it belongs to the NP language class and finding a polynomial transformation from Vertex Cover (a known NP-complete language) to Hamiltonian Circuit.

2.2 Vertex Cover Problem

A Vertex Cover is a subset of vertices from a graph that contains at least one endpoint of every edge of the graph, the problem of finding a Minimum Vertex Cover is a classical optimization problem and its NP-Complete.

The formal definition of the problem would be:

Given a Graph $G = (V, E)$ and an integer $K \in \mathbb{N}$

The Vertex Cover $V' \subseteq V : uv \in E \Rightarrow (u \in V' \vee v \in V')$

Can we find a vertex cover with $|V'| = K$?

We know the Vertex Cover problem is NP-Complete because its "easy" to find a polynomial time algorithm to solve it in a non-deterministic turing machine, and it has been proven that the Vertex Cover problem is NP-Complete, it has been found a polynomial transformation from the 3-SAT problem to Vertex Cover.

2.3 Hamiltonian Circuit Problem

The Hamiltonian Circuit problem consists of finding a cycle in a graph where every node of the graph appears only once.

The formal definition of the problem is:

Given a Graph $G = (V, E)$
 Does G contain a Hamiltonian circuit?
 That is, an ordering $\langle v_1, v_2, \dots, v_n \rangle$ of the vertices of G
 where $n = |V| : (\{v_n, v_1\} \in E) \wedge (\{v_i, v_{i+1}\} \in E) \quad \forall i, 1 \leq i < n$

2.4 Proving Hamiltonian Circuit \in NP

To prove that the Hamiltonian Circuit problem is in NP we will be using the Certifier-Certificate based definition of NP. This way of proving if a problem is in NP relies on having a certificate (solution to the problem) and a certifier (a way of checking if the certificate is a solution to the problem) in polynomial time, if so, the problem is in NP.

Given a 'certificate' for the Hamiltonian Circuit problem, this is a sequence of vertices forming the hamiltonian cycle, we will be able to verify the certificate by checking that all of the vertices belong to the graph and each pair of vertices in the certificate are adjacent:

Listing 1: Polynomial verifier for Hamiltonian cycle

```

for {u, v} in V' where v = u+1
  if E does not contain {u, v}
    Solution is incorrect
  endif
endfor
Solution is correct
  
```

2.5 Polynomial Transformation

The polynomial transformation from Vertex Cover to Hamiltonian Circuit is based in the construction of components, specifically, we will have two types of components, the selectors and the more complex cover-testing components.

2.5.1 Selectors

The selectors is the simplest set of components we will need to do the transformation, its just a set of vertices $S = \{a_1, a_2, \dots, a_k\}$ with $|S| = K$ where K is the size of the vertex cover.

2.5.2 Cover-Testing components

We will add a Cover-Testing component for every arc in the original Vertex Cover Graph. The purpose of these Cover-Testing components is to ensure that at least one of the vertices of the original arc is in the Vertex Cover.

For each $e = u, v \in E$ of the Vertex Cover graph a cover testing component has:

A vertex set $V'_e = \{(u, e, i), (v, e, i) : 1 \leq i \leq 6\}$ with $|V'_e| = 12$
 And edges $E'_e = \{ \{(u, e, i), (u, e, i+1)\} \{ (v, e, i), (v, e, i+1) \} : 1 \leq i \leq 5 \}$
 $\cup \{ \{(u, e, 3)\}, \{(v, e, 1)\}, \{(v, e, 3)\}, \{(u, e, 1)\} \}$
 $\cup \{ \{(u, e, 6)\}, \{(v, e, 4)\}, \{(v, e, 6)\}, \{(u, e, 4)\} \}$
 with $|E'_e| = 14$

This constructions makes sure that the only vertices that will be involved in any external edges are $(u, e, 1), (v, e, 1), (u, e, 6)$ and $(v, e, 6)$. This implies that any Hamiltonian circuit of G' will only be able to pass through the edges of every vertex cover in one of the three following configurations:

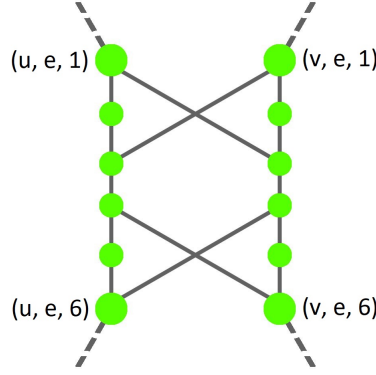


Figure 1: Cover-Testing construction

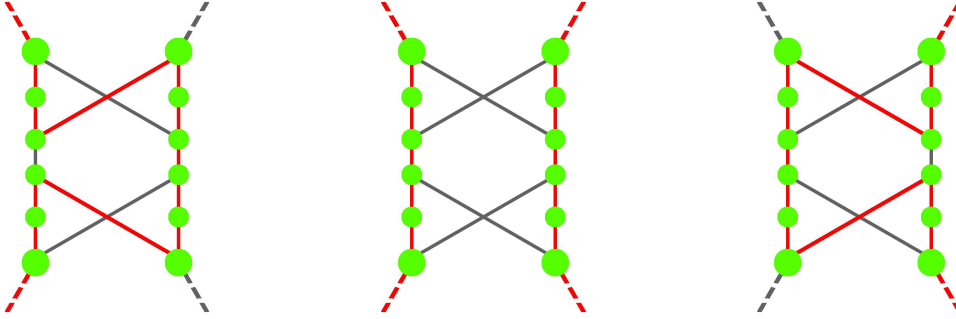


Figure 2: The three possible configurations, of the Hamiltonian Circuit of a Cover Testing for an edge $e = u, v$. From left to right, they correspond to the cases where u is part of the vertex cover, where u and v are part of the vertex cover and where only v is part of the vertex cover.

2.5.3 Joining the components

After placing one Cover-Testing in every arc of the original Graph, we'll need additional arcs to connect pairs of Cover-Testing components or a selector with a Cover-Testing component. To do this, for each vertex $v \in V$ we will order all of the incident edges on v and all of the Cover-Testing components corresponding to those edges are joined together adding the following edges:

$$E'_v = \{ \{ (v, e_{v[i]6}), (v, e_{v[i+1]1}) \} : 1 \leq i < \deg(v) \} \text{ Where } \deg(v) \text{ is the degree of } v$$

This additional edges will form paths between Cover-Testing components, and the final connecting edges in G' will join the first and last vertices from each of these paths to the selector vertices $S = \{a_1, a_2, \dots, a_k\}$

$$E'' = \{ \{ a_i, (v, e_{v[1]1}) \}, \{ a_i, (v, e_{v[\deg(v)]6}) \} : 1 \leq i \leq K, v \in V \}$$

2.5.4 Final demonstration

The resulting graph $G = (V', E')$ of the transformation has:

$$V' = \{ a_i : 1 \leq i \leq K \} \cup \left(\bigcup_{e \in E} V'_e \right)$$

and

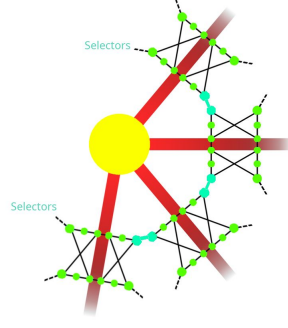


Figure 3: Example of the path made by connecting all the Cover-Testing components from an original node.

$$E' = \left(\bigcup_{e \in E} E'_e \right) \cup \left(\bigcup_{e \in E} E'_v \right) \cup E''$$

Its easy to see how the transformation has polynomial complexity:

Listing 2: Polynomial verifier for Hamiltonian cycle

Generate $12* E $ Cover-Testing vertices +	
K selector vertices	$O(n)$
Generate the $14* E $ Cover-Testing internal edges +	
$2* E -1$ connections between components +	
$k*2* E $ selector edges	$O(nk)$

Since $k \leq |V|$ in the worst case scenario $k = |V| = n$ so the complexity is $O(n^2)$

Finally, we have built a Graph $G' = (V', E')$ such as if a solution to the Vertex Cover Problem with size K exists on graph G , then there has to exist a Hamiltonian Circuit in G' .

2.6 Sources

Computers and Intractability: A Guide to the Theory of NP-Completeness - Garey, Johnson

[Hamiltonian path on Wikipedia](#)

[Proof that Hamiltonian Cycle is NP-Complete by GeeksForGeeks: NP on Wikipedia](#)