

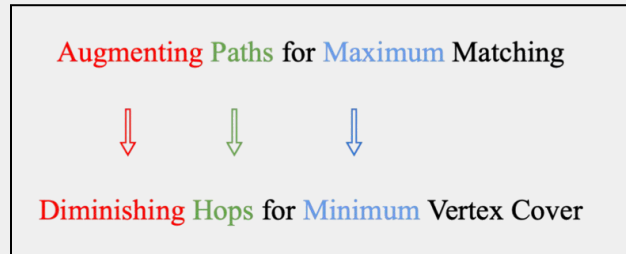
On the Computational Complexity of the Vertex Cover Problem on Cubic Bridgeless Graphs

TL;DR

The vertex cover problem on cubic bridgeless graphs is an NP-complete problem that is also in P.

A Core Contribution:

- **Novel Data Structure:** Represents table.
- **Novel Concept:** Diminishing hops (analogous to augmenting paths in maximum matching).
- **Graph-Theoretic Theorem:** Bridges diminishing hops and minimum vertex cover (analogous to Berge's Theorem).
- **Algorithmic Result:** Deploys diminishing hops to find a minimum vertex cover (analogous to the Blossom Algorithm using augmenting paths to find a maximum matching using Berge's Theorem).



The Cook-Levin Theorem, directly and indirectly, ushered in an era of one of the deepest scientific inquiries in the world of computer science and mathematics. One early example of a direct inquiry is of Karp whose seminal result showed that 21 computational problems are NP-complete. One of those 21 problems is the vertex cover problem, the topic of our discussion.

What is the vertex cover problem?

Given an unweighted undirected graph, the vertex cover of the graph is a set of vertices that includes at least one endpoint of every edge of the graph. The corresponding computational problem of finding the minimum-size vertex cover is the vertex cover problem and it is known to be NP-hard. Here, we focus on the (decision version of the) vertex cover problem on cubic bridgeless graphs (VCCBG).

Where does this focus lead to?

This focus leads to a multi-part study on understanding the computational complexity of VCCBG where it was discovered that (a) VCCBG is NP-complete and (b) VCCBG is in P.

How so?

Part I - It proves that VCCBG is NP-complete by showing that (i) VCCBG is in NP and (ii) VCCBG is NP-hard via a polynomial-time reduction from a known NP-hard problem, namely, the vertex cover problem on cubic graphs.

Part II - In this core contribution, an unconditional deterministic polynomial-time exact algorithm for VCCBG is discovered. The algorithm is divided into three phases:

- Phase I uses the Blossom Algorithm [Edmonds, 1965] to find a maximum matching, which is a perfect matching when using a cubic bridgeless graph [Petersen, 1891].
- Phase II constructs a tree from the given graph by doing a breadth-first search of lexicographically sorted vertices. It then augments the 2-approximation algorithm of the vertex cover problem to populate a novel data structure called the "represents table" (RT), which stores the information of each endpoint and the neighbors of each endpoint.
- Phase III begins with the removal of some endpoints from the RT such that the frozen endpoints still form a vertex cover. Then, a local-to-global property called the "diminishing hops" is introduced. Specifically, just like Berge's Theorem [Berge, 1957] established the relation between the absence of an augmenting path (local property) and a maximum matching (global maximum), this phase establishes the relation between the absence of a diminishing hop and a minimum vertex cover (MVC). Consequently, this relation forms the basis of Phase III of the algorithm, analogous to Berge's Theorem forming the basis of the Blossom Algorithm.

Part III - It provides an alternative proof to show that VCCBG is NP-complete using proof by restriction (Section B). It follows a simple chain of arguments that arise from two facts:

- vertex cover problem on 2-connected cubic planar graphs is NP-complete [Mohar, 2001]
- every 2-connected graph is bridgeless [Whitney, 1932]

Hence, the NP-complete vertex cover problem on 2-connected cubic planar graphs is equivalent to the vertex cover problem on bridgeless cubic planar graphs. Next, the vertex cover problem on bridgeless cubic planar graphs is a subset of the vertex cover problem on bridgeless cubic graphs. Therefore, VCCBG is also NP-complete.

Part IV - It provides a simpler alternative algorithm to prove VCCBG is in P (Section C):

- introduce the concepts of alternating bipartite graph and diminishing bipartite graphs for vertex cover (analogous to alternating and augmenting paths for matching)
- bridge diminishing bipartite graphs and MVC (analogous to Berge's Theorem)
- utilize diminishing bipartite graphs to find a MVC (analogous to the Blossom Algorithm)

So what?

There are four sets of proofs demonstrating $P = NP$.

Even then, is it possible that for a *given* problem, there exists a gap between the time taken to verify a given solution and the time taken to solve the problem, under a few standard conditions? After all, at the cost of mathematical precision, one may wonder if even Beethoven found composing a tune "harder" than "verifying" one, and if even Ramanujan found discovering a proof "harder" than verifying one?

Resources:

Link to the paper: <https://kunalrelia.github.io/img/VCCBG.pdf>

Link to Lean formalization: <https://github.com/KunalRelia/VCCBG>

Part I, Part II, Part III, and key results from Part IV have been formalized.

Comments are highly appreciated. You may use [this](#) form to share comments.

Email: kunal.relia91@gmail.com / krelia@nyu.edu