Cycles in Graphs

Brent Kirkpatrick

April 4, 2022

© 2022 Intrepid Net Computing

Intrepid Net Computing

www.intrepidnetcomputing.com

Document Revision History

April 4, 2022 Published original on www.intrepidnetcomputing.com

Abstract

Studying cycles in a graph is done by composing simple cycles. Simple cycles are either node simple or edge simple. The main result is regarding edge simple cycles. Edge simple cycles can be enumerated in polynomial time.

The study of cycles in graphs begins with the work of Euler. An Eulerian cycle is an edge-complete cycle that visits each edge exactly once. A Hamiltonian cycle is a node cycle that visits each node in the graph exactly once.

Graphs are studied in algebra, in operations research, in machine learning, in matrix theory, and in many other areas. The most famous cycle is the traveling sales person cycle which is a solution to the NP-complete problem in graph theory that is dubbed the traveling sales person [1] problem. In this formalism, we have a graph of cities with edges being routes between the cities. Each edge has a weight indicating the cost of travel. The problem is to find the minimum weight tour that visits all the cities that the traveling sales person must visit. This tour begins and ends at the same node, but may not include every node or edge of the graph.

Background

A cycle is simple if it contains each edge at most once. We list the nodes of a simple cycle in the order that the edges appear in the cycle. A cycle is assumed to begin and end at the same node. See Figure 1 for an example.

The problem of determining whether a graph has a hamiltonian cycle is NP-complete [2, 3]. The problem of determining whether a graph has a eulierian cycle is polynomial [4].

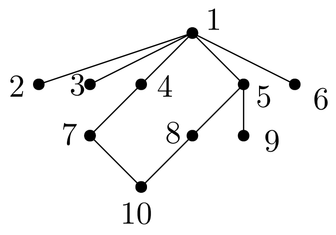

Figure 1: **Simple Cycle.** The notable simple cycle in this example is cycle $\{1, 4, 7, 10, 8, 5\}$. This is an undirected cycle.

In this context of exponential time algorithms for finding some cycles, our question is one of finding any cycles quickly. A clique in a graph is an arrangement of nodes and edges in a complete subgraph, where every permutation of the nodes in the subgraph is a cycle. In a clique there is a linear-time algorithm for finding a cycle. However, finding a max clique is a hard problem.

Simple cycles compose to form other cycles. The composition of two cycles might be simple or not simple. For example, Figure 2 shows two simple cycles. The cycle $\{1, 4, 7, 10, 8, 5\}$ can be composed with the cycle $\{10, 11, 13, 12\}$ to obtain another cycle, simple or not. For example, a not simple cycle is $\{1, 4, 7, 10, 11, 13, 12, 10, 7, 4\}$.

Our problem is to find all the simple edge-disjoint cycles in a graph. There is a polynomial time algorithm for this. This algorithm makes use of a result for graph isomphism [5].

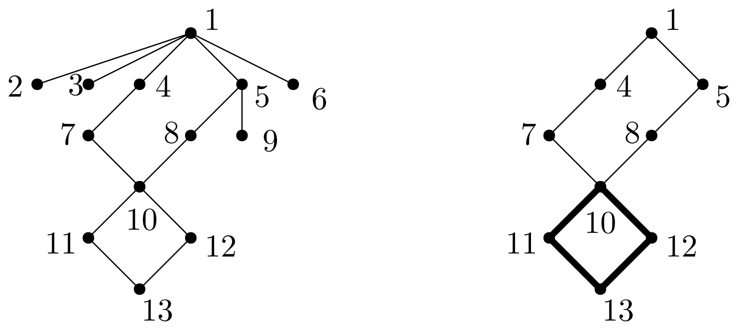

Figure 2: **Cycle Composition.** (left) The undirected graph. (right) Two undirected cycles. One cycle has light edges and one cycle has heavy edges. The cycle composed of both cycles is also a simple cycle.

Algorithm

Let $G = (V, E)$ be an undirected graph with labled nodes V and edges E. We direct the edges by finding the high degree nodes and using a breadth-first search to direct edges away from the high degree nodes. This is a deterministic way to direct the edges.

For each node in the graph, we obtain the descendent set. With the d-splits of the graph, we discuss the algorithm.

For each source node, we compare the child nodes to find the child d-splits that contain a common descendent. Define a **same** descendent as a descendent that appears in the decendent set of more than one child of a source node. When we find a same descendent, there is a path from the source to the same descendent through each of the child nodes.

Theorem 1. *For every same descendent, there is a simple cycle.*

Proof. We find the cycle by using the directed paths in our directed graph. The same descendent has a source node and two child nodes. Now we follow the directed path from the source through a child node to the descendent. We do this twice. We remove all leaf paths. This gives us a simple cycle in the undirected graph. □

This algorithm continues by traversing all directed paths from any source with child nodes having a same descendent to the same descendent. All these directed paths constitute the simple cycles of the undirected graph.

Theorem 2. *The algorithm given lists every edge in every simple cycle of more than one edge.*

Proof. Assume that there is a cycle edge not listed in a simple cycle. Either the edge would not be traversed or it would not be on a path with a same descendent. If the edge is not traversed, there is not a descendent set with a same descendent. If there is no child with a same descendent, there is not a cycle. This is a proof by contradiction. □

Each simple cycle is a tour through the nodes in the cycle. Some of these simple cycles are also node simple. The undirected graph is hamiltonian if and only if there is a node simple cycle for the whole graph.

The main result is that cycles compose. The Euler tour is made by composition.

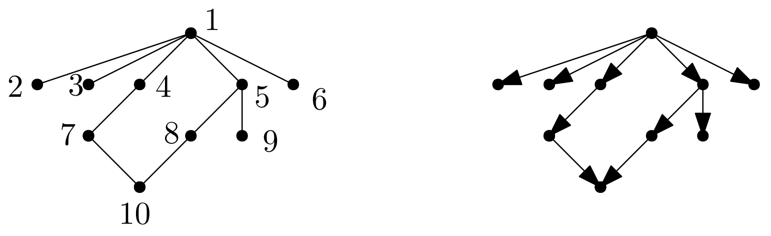

Figure 3: **Example.** (left) The undirected graph. (right) The edge directions are given. The algorithm selects two paths using the d-splits. The first path is $\{1, 4, 7, 10\}$ and the second path is $\{1, 5, 8, 10\}$.

Conclusions

This paper gives a new algorithm for finding all the simple cycles of an undirected graph. This algorithm is polynomial and produces all the simple cycles of more than one edge. The simple cycles can be composed. This also suggests a new algorithm for finding the eulerian tour of a graph by composing simple cycles.

References

[1] A. Turing. The traveling sales person.

[2] R. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 1972.

[3] T. Cormen., C. Leiserson, R. Riverst, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[4] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer-Verlag, New York, 2001.

[5] B. Kirkpatrick. A polynomial-time algorithm for graph isomorphism. *TIPS*, 2016.