4-13-2018

Graph Pebbling: Doppelgangers and Lemke Graphs

Mark Powers

Graph pebbling algorithms and Lemke graph construction

Mark P. Powers, Dr. Charles A. Cusack, and Dr. Airat Bekmetjev

Hope COLLEGE

Graphs

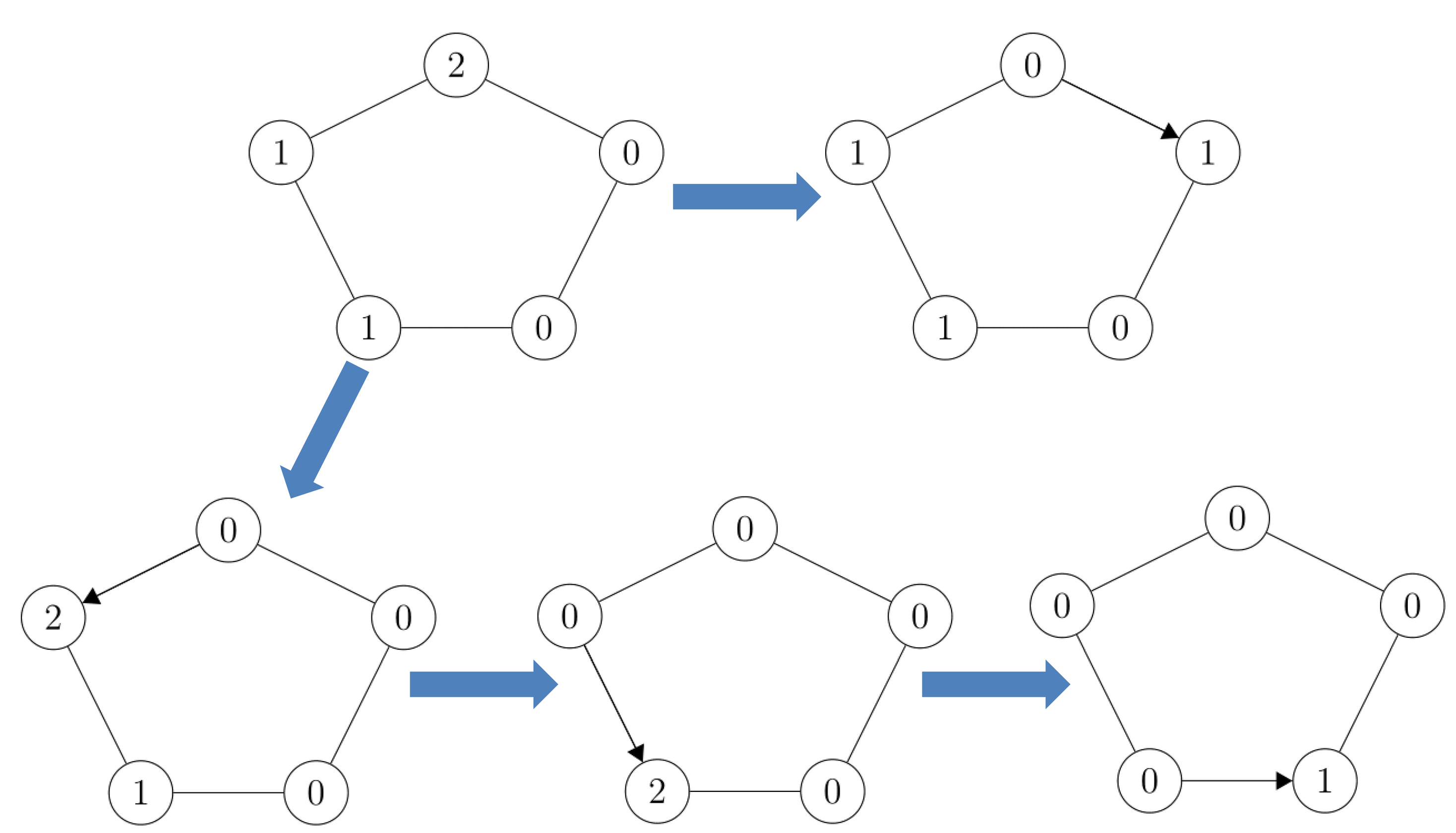
A graph is a structure composed of nodes (or *vertices*) that are connected with edges.

- The set of vertices of a graph *G* is denoted *V(G).*
- The *diameter* of a graph *G*, written *diam(G),* is the maximum length of all shortest paths between all vertices in *G*.
- Two vertices are *adjacent* (or *neighbors*) if they are connected with an edge.
- A graph is *complete* if there are edges between all pairs of vertices.

Graph Pebbling

Graph pebbling is a game played on a graph.

- A *configuration* places a non-negative number of pebbles on vertices of a graph.
- A *move* removes two pebbles from one vertex and places one pebble on an adjacent vertex.
- A vertex is *reachable* under a configuration if after a sequence of moves, at least one pebble can be moved to that vertex.
- A configuration is *solvable* if all vertices are reachable.



- The *pebbling number* of a graph *G*, written as $\pi(G)$, is the smallest number so that any configuration of at least $\pi(G)$ pebbles is solvable on *G*.
    - $\pi(G) \geq |V(G)|$
    - $\pi(G) \geq 2^{diam(G)}$
- The *two-pebbling property* of a graph *G* holds if any configuration of more than $2\pi(G) - q$ pebbles, where *q* is the number of vertices with pebbles, allows for moving two pebbles to any vertex
    - Graphs that do not have the *two-pebbling property* are called *Lemke* graphs

Algorithms

To determine solvability for a graph, we first used four nondeterministic algorithms. These algorithms don't always determine if a configuration is solvable, but they are very fast.

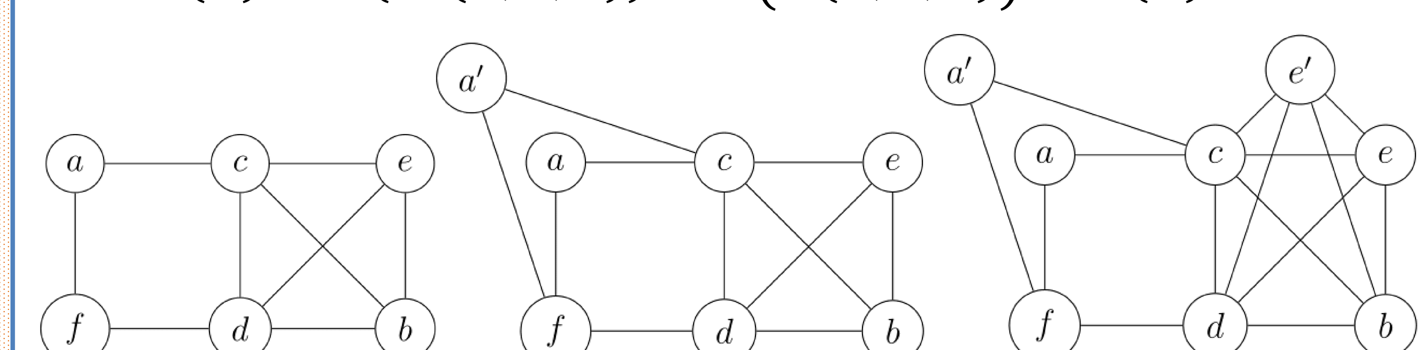
- *Domination Check*
    - All vertices within distance k from a vertex with 2^k pebbles are marked reachable.
- *Shortest Path*
    - For each vertex, all pebbles are moved along a shortest path to that vertex.
- *Shortest Pebble Path*
    - Similar to the *shortest path* algorithm, but the distance between two vertices takes into account pebbles along a path, and considers these shorter.
- *Weight Function*
    - Given a vertex *r*, we assign the weight of a pebble on *v* to be $2^{-dist(v,r)}$. If the sum of this for all vertices and pebbles is less than one, then *r* cannot be reachable.

Doppelgangers

Two vertices are *doppelgangers* if they have the same neighbors. We define $D(G, v, i)$ to be the graph *G* with *i* additional doppelgangers (called *doppels*) of the vertex *v*. We define $D'(G, v, i)$ to be the graph $D(G, v, i)$ where all doppels and *v* are connected.

For a graph *G* with more than 3 vertices, let $k \geq 1$ and $v \in V(G)$

$$\pi(G) \leq \pi(D'(G, v, k)) = \pi\big(D(G, v, k)\big) \leq \pi(G) + k.$$



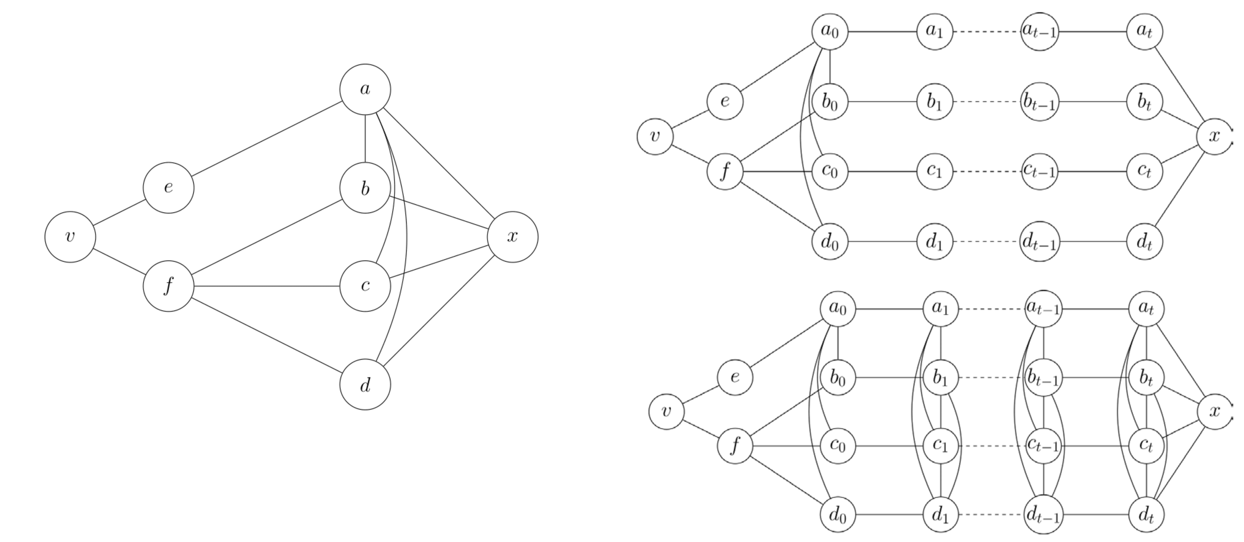
Graham's Conjecture

The cartesian product of two graphs *G* and *H* replaces each vertex in *G* with a copy of *H*. Graham's conjecture states that for two graphs *G* and *H*, $\pi(G \square H) \leq \pi(G)\pi(H)$

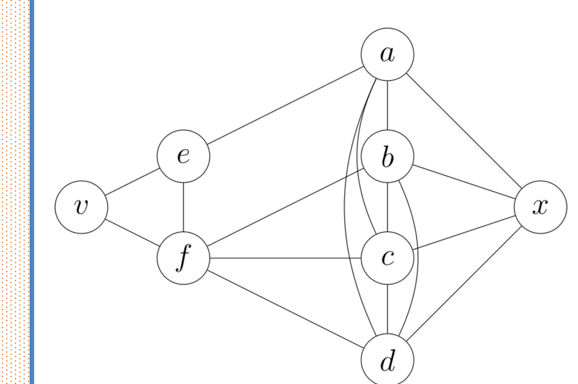
Many of these results rely on the two-pebbling property of a graph. Lemke graphs are suspected to be involved if there is a counterexample, especially the product of two Lemke graphs. Understanding the structure of Lemke graphs could prove useful in doing this.

Lemke Graphs

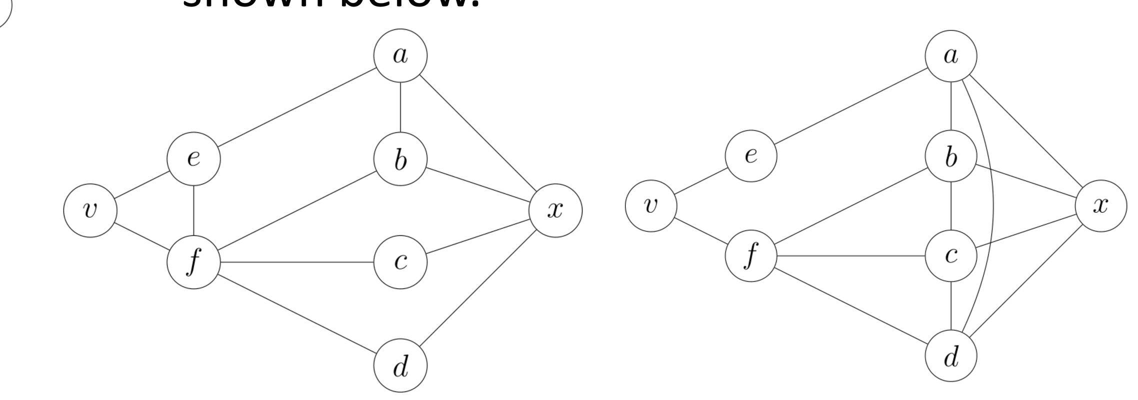
Originally, there was one known Lemke graph. This graph was modified to form two infinite sequences of Lemke graphs.



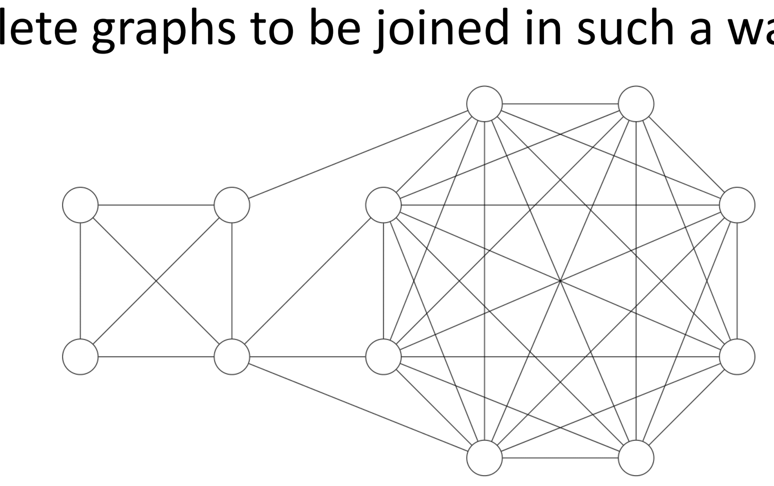
Using our algorithms on all graphs through nine vertices, we found that there is one maximal Lemke graph of eight vertices, where all other eight vertex Lemke graphs are subgraphs of it.



Additionally, there are three minimal Lemke graphs on eight vertices. One of these is the original Lemke graph, the other two are shown below.



Using doppelgangers, we are able to construct Lemke graphs by adding any number of doppels to the vertices on the left and right. This allows for any two complete graphs to be joined in such a way that creates a Lemke graph.



Acknowledgements

We would like to thank previous students Aaron Green and Cole Watson for their work on this project.