Construction of Eulerian Trails in Large Graphs

Stefan Muller
Carnegie Mellon University

Ben Blum
Carnegie Mellon University

Abstract

We went on a long walk.

1. Introduction

We begin this paper, as is the case with most dry, theoretical algorithms papers, with some flavor text designed to convince you to care about the algorithm presented in this paper. Here goes.

Suppose, hypothetically, you are an academic researcher who enjoys taking occasional walks during the day. Suppose further that you live in a city with highly variable weather, so you want to take a long walk indoors. You could walk around in circles, but then, totally hypothetically, the undergrads sitting in the lounge near your office might think you're crazy if they keep seeing you walk by. So you want to go for a long walk without covering the same stretch of hallway twice. Crossing your path is fine.

It turns out that, like every other problem, this can be reduced to a question about graphs and, also like every other problem, this one has already been studied by Euler and it's called an Eulerian trail. You could look this up, but we'll save you the trouble and remind you that Euler conjectured that a graph has an Eulerian cycle (which is like an Eulerian trail but it starts and ends in the same place so you don't have to look like an idiot when you retrace your path back to your office) exists in a graph if and only if every vertex in the graph has even degree. This claim was tested and confirmed by Hanneman and Blvm [2].

As if that wasn't enough, Euler also conjectured that if all but two vertices of the graph have even degree, then there's an Eulerian trail from one to the other. In this paper, we empirically test this claim by constructing an Eulerian trail on a large graph.

2. Large Graphs

Since big data is all the rage [3], we obviously want to construct an Eulerian trail on a large graph. The large graph we use is shown in Figure 1. Due to the size of the graph, we do not label each node but rather label "regions" consisting of at least two nodes each. Region names are not meaningful. This may not seem like a large graph in terms of the number of nodes or edges. The diameter of the graph, on the other hand, is approximately 350m, making it a relatively large graph, though admittedly not as large as the graph on which Hanneman and Blvm ran their experiments.

A simple counting argument[1] shows that all of the vertices but two, G_2 and R_2, have even degree. That means that an Eulerian trail of the graph should exist starting at G_2 and ending at R_2. In the rest of the paper, we constructively prove this.

3. Proof

We find the Eulerian trail of the graph using Fleury's algorithm:

1. Start at a vertex of odd degree.
2. While there are edges left:
   (a) Find a bridge, that is, an edge that would not disconnect the graph if deleted.
   (b) Follow it.
   (c) If all the edges would disconnect the graph, just follow one of them, OK?
3. You're done.

Fleury's algorithm traverses a graph with E edges in $O(E)$ time. This analysis has been criticized because it ignores the time required to find bridges in step 2a [1]. Fortunately, we have an $O(1)$ algorithm for finding bridges. They look like this:

We performed the algorithm on the graph of Figure 1, starting at vertex G_2. Our results are below:

[1] Just count.

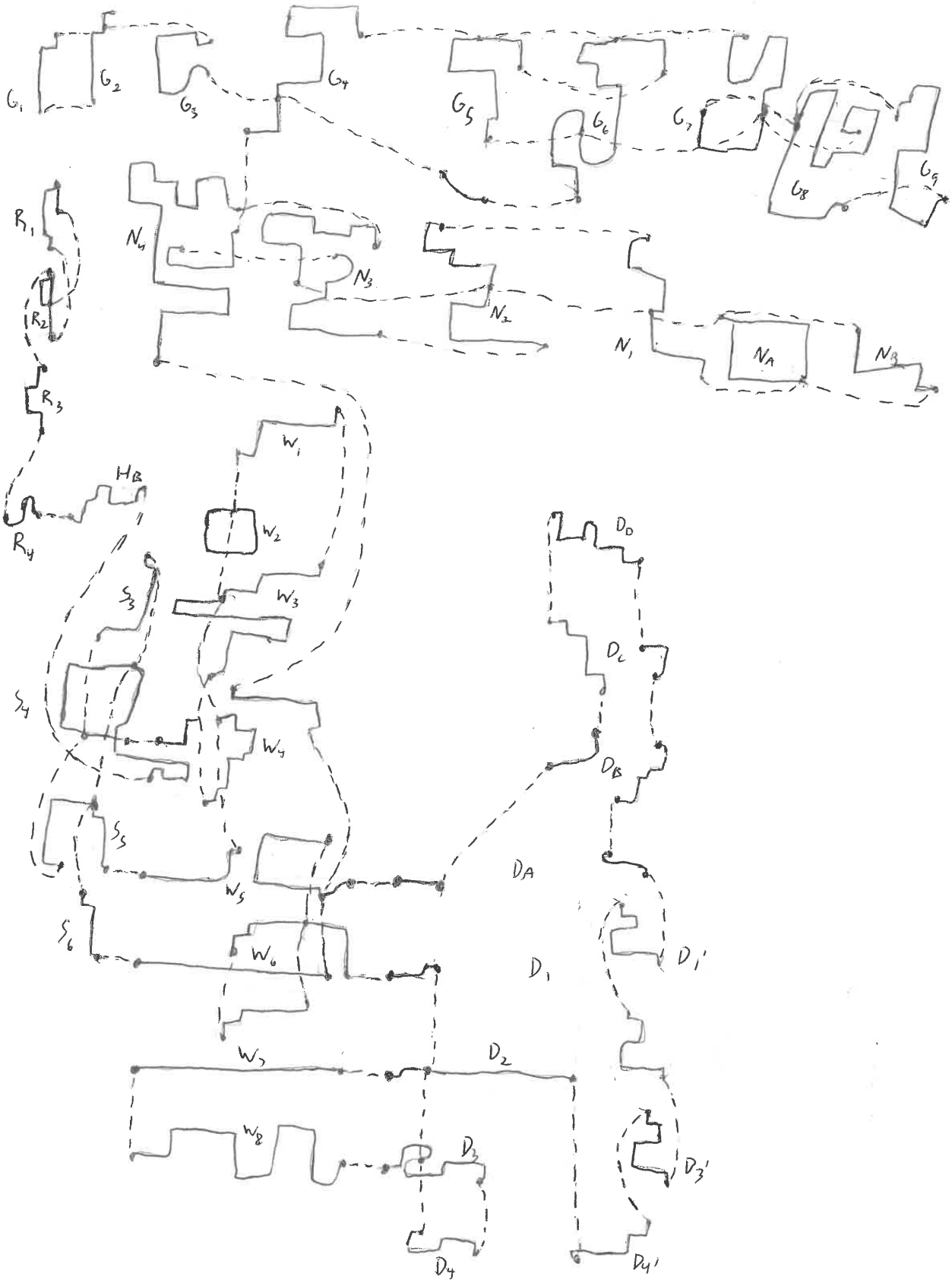

Figure 1. The graph. TikZ is hard [**?**], OK?

Running time	8100s
# of steps	7802
Approx. distance	5461m

4. Conclusion

This initial study has found an Eulerian trail in a large graph. As additional infusions of money continue to expand this graph, we expect that more studies of this kind will become possible.

References

[1] Eulerian path. `https://en.wikipedia.org/wiki/Eulerian_path#Fleury's_algorithm`.

[2] Greg Hanneman and Benjamin Blum. A constructive solution to the k onigs-pittsburgh problem. In Proceedings of the ninth SIGBOVIK, pages 21–24, 2015.

[3] Keith A. Maki. A modular approach to state-of-the-art big data visualization. In Proceedings of the eleventh SIGBOVIK, pages 172–175, 2017.