New Challenges in Planar Emulators

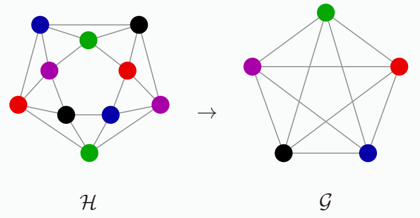
Faculty of Informatics, Masaryk University
Brno, Czech Republic

Martin Derka
mderka@mail.muni.cz

David R. Cheriton Scool of Computer Science
University of Waterloo, Waterloo, ON, Canada

Problem

Definition 1 (Planar Emulator)
Graph $\mathcal{G}$ has a planar emulator $\mathcal{H}$ if $\mathcal{H}$ is a finite planar graph and there exists a homomorphism from $\mathcal{H}$ onto $\mathcal{G}$ that is locally surjective.



Graph $\mathcal{G}$ on the right side is known not to have a drawing in the plane without edge crossings (Kuratowski, 1930). However, we can construct graph $\mathcal{H}$ which has the same inner structure as $\mathcal{G}$ (i.e. every vertex has all the neighbours as in $\mathcal{G}$) and it can be easily drawn in the plane. We call $\mathcal{H}$ a planar emulator for $\mathcal{G}$.

The following are two main questions in the field of finite planar emulators:

When does a finite planar emulator for a given graph exist?
What is the class of graphs with planar emulators?

Conjecture 2 (Fellows 1988, Kitakubo 1992)
The class of graphs with a finite planar emulators is equal to the class of graphs with a projective embedding.

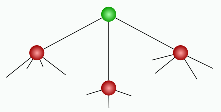
For almost 20 years, there has been no progress in this field. Conjecture 2 was believed to be true until two emulators for two non-projective graphs, comonly denoted by $K_{4,5} - 4K_2$ and $K_{1,2,2,2}$, were surprisingly published in 2008 (Rieck, Yamashita). This shows that **non-projective graphs with planar emulators exist**, but we do not know anything about them. **The goal of our work is to provide characterization of all such graphs.**

Motivation

Even though the problem of finite planar emulators is formulated purely theoretically, it can have many practical consequences. Planar graphs are nice because many problems, which are hard on graphs in general, can be solved quickly (in a polynomial time) if the graph is planar. A proper understanding of the concept of planar emulation can allow us to extend those fast polynomial algorithms to other graphs that are non-planar themselves, but have a finite planar emulator.

Our Approach

In our work, we prove that if a non-projective graph $\mathcal{G}$ has a finite planar emulator, it can be trivially derived from a non-projective graph that is internally 4-connected (see below). This means that the only interesting graphs are the internally 4-connected non-projective graphs that do not violate some other restrictions that must be met by a graph with a finite planar emulator.



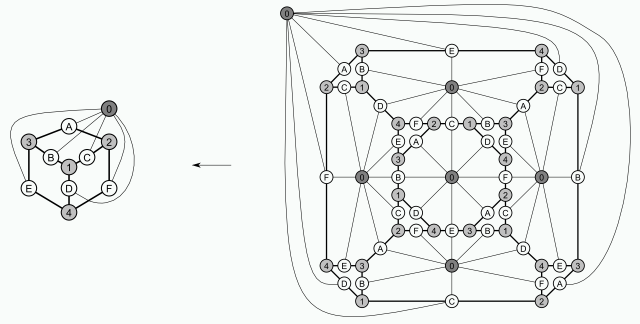
In an internally 4-connected graph, every vertex with precisely 3 neighbours must satisfy the condition that those 3 neighbours are not connected by an edge (i.e. no edge between the red vertices).

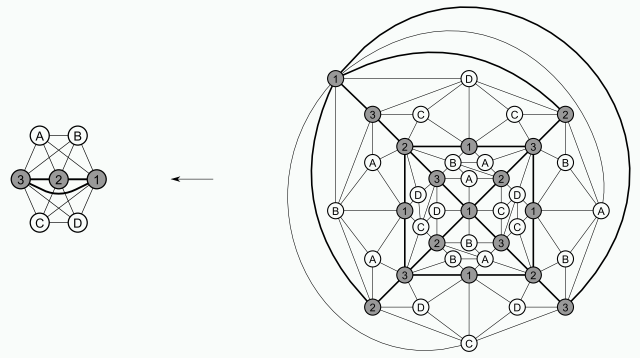
The class of internally 4-connected non-projective graphs can be systematically generated. We implemented a computational framework in order to enumerate all such graphs to show that only a small number of them can have a finite planar emulator. Even though all the graphs of our interest can be systematically generated, it is not an easy task as the number of them is enormous.

We conducted many experiments within which we explored all the known ways of generating internally 4-connected graphs. Our first computations required hundreds of hours of computations employing supercomputers and grids. Most of our computations were conducted on 128 core Intel Xeon X7560 @ 2.27 GHz at the Faculty of Informatics, Masaryk University, and using the computational grid Metacentrum.

Getting a better understanding of how those generating methods apply to our class of graphs, we were able to pick the best method and suggest further optimization, which now allows us to conduct all the necessary experiments on a regular personal computer, e.g. our 6-year-old Intel Core 2 Duo @ 2.0 GHz.

More Examples of Emulators





Results

(1) We proved that, up to a limited number of exceptions, there are only a few graphs that contradict Fellows' characterization of graphs with finite planar emulators.

(2) We discovered some interesting structural anomalies in the class of internally 4-connected graphs that we call "violating cycles". These structures are of an extreme interest with respect to other problems.

(3) Proof of our results includes a computational part. We provided a clear description of our computational framework and optimized algorithms. This allows easy independent verification of our results in a commonly accessible setup.

(4) We outlined a computational approach to traversing internally 4-connected graphs. This approach can be applied to other problems in graph theory.