Description. In this project, we will investigate a topic that has fascinated geometers for many years: the flip graph of triangulations. Its study dates back to a paper by Wagner from 1936 [5].

Triangulations are planar graphs where every face is a triangle. For the purpose of this project, we will consider abstract unlabeled (i.e. isomorphism classes of) triangulations. Every edge \(\{u, v\}\) in a triangulation \(G\) has a face on either side, say \(uv_s\) and \(v_ut\). If \(\{s, t\}\) is not already present in \(G\), we say that \(\{u, v\}\) is flippable: we can delete \(\{u, v\}\) and replace it by \(\{s, t\}\). We call this operation a flip. The flip graph \(\mathcal{F}_n = (V, E)\) is the graph where \(V\) is the set of triangulations on \(n\) vertices and \(\{G_1, G_2\} \in E\) if and only if \(G_1\) can be transformed into \(G_2\) by a single flip. Figure 1 depicts a series of flips in a combinatorial triangulation, which corresponds to a path in the flip graph. The diameter \(|\mathcal{F}|\) of \(\mathcal{F}\) is the maximum length of a shortest path in \(\mathcal{F}\). It is known that \(|\mathcal{F}_n|\) is between \(2n - 15\) [4] and \(5.2n - 24.4\) [1]. A detailed survey on the history of the problem is available at [2].

Goal. As a starting point, you will write a computer program to compute flip graphs for small \(n\). There already exist tools for generating all maximal planar graphs [3]. Since the number of maximal planar graphs grows quickly with \(n\), part of the challenge is to come up with some smart ideas to push the boundaries of what you can compute. These graphs may give insight on how to improve the existing bounds of \(|\mathcal{F}|\). From there, we can continue research in a more theoretical direction, depending your preference, which is the ultimate goal.

Prerequisites. You should be familiar with C++. Ideally, you have some knowledge of computational geometry (e.g. from our Computational Geometry or Geometry: Combinatorics & Algorithms courses).

References.


Contact. Vincent Kusters <vincent.kusters@inf.ethz.ch> and Michael Hoffmann <hoffmann@inf.ethz.ch>