Graph separation problems from the perspective of parameterized complexity

1 The concept and objectives

Assuming P $\neq$ NP, in order to solve NP-hard problems, we have to use superpolynomial algorithms. *Prameterized complexity* introduces a concept of dealing with those problems, by considering different measures of the hardness (in contrast to the size of the input), and constructing algorithms that are efficient when the hardness is relatively small. A problem is *fixed parameter tractable* (FPT), if there exists an algorithm solving it in time $f(k) \cdot |x|^{\mathcal{O}(1)}$, where x is an instance, k is an integer being the aforementioned measure of the hardness, called *the parameter* and f is some computable function.

The study of cuts and flows is one of the most widely studied fields in combinatorial optimization. A classical text-book example of the problem is *Min Cut*, in which for a given graph G, we are looking for a minimum set of edges that removal disconnects G. While Min Cut has polynomial algorithms, often even a slight modification of it, becomes intractable, e.g. when we seek for a cut that splits G into two equal parts, so-called *Min Bisection*.

The main aim of the project is to look for efficient FPT algorithms for NP-hard graph separation problems, as well as to prove that we cannot solve some of them faster than currently is known. It contains the following three parts.

In the first part, the goal is to improve parameterized algorithms for two classical problems, i.e. *Node Multiway Cut*, and Min Bisection, with the parameter being the solution size.

The vertex version of a cut problem very often turns out to be more challenging. In Node Multiway Cut, for a graph G, a set of terminal vertices T, and a parameter k, we are looking for a set S of at most k non-terminal vertices, that removal disconnects T, i.e. there is no path between any $t_1, t_2 \in T$ in $G - S$. The best known algorithm for Node Multiway Cut has complexity 2^k [1] (by Cygan et al.), while an edge version, i.e. Edge Multiway Cut poses 1.84^k algorithm (by Cao et al.). We will investigate whether it is possible to break constant 2 in this problem as well.

Min Bisection has been proven to be FPT by Cygan et al. [STOC 2014], and the authors presented $2^{\mathcal{O}(k^3)}$ algorithm. They also proposed a special kind of *tree decomposition*, which *bags* have highly connected structure, but can have arbitrary sizes. We will try to construct faster algorithm for the decomposition, that can lead to an algorithm for Min Bisection.

In the second part, we will consider so-called *square root phenomenon* in subexponential algorithms. Core problems that have $2^{\mathcal{O}(\sqrt{k})}$ algorithms are problems on planar graphs. There are only a few other problems that have the same upper bound. In particular, they are completion problems, i.e. *Chordal Completion*, *(Proper) Interval Completion*, *Trivially Perfect Completion*. Understanding the tightness of those bounds is much harder than in planar graphs.

In Bliznets et al. [SODA 2016], we have proved that unless *Exponential Time Hypothesis* fails, there is no $2^{o(\sqrt[4]{k})}$ algorithms for those problems. We have also introduced a plausible hypothesis implying the desirable lower bounds. The hypothesis assumes the subexponential hardness of constant approximation of Min Bisection. We would like to push the project further, and connect the stated hypothesis with similar Feige's hypothesis, about the 3-SAT hardness on average.

The aim of the last part, is to improve algorithms for cut problems in the classical sense of exact algorithms, i.e. put the hardness measure to be the number vertices of the graph.

We will consider two problems with long FPT literature, for which breaking the brute force $2^{|V|}$ bound is not that easy, i.e. *Multicut* and *Directed Feedback Arc Set*. We will also work on creating an algorithm for Min Bisection. Surprisingly, it has no direct history with those exact algorithms.

2 Reasons for choosing the research topic

The impact of the project on the development of science is self explanatory. People try to tackle NP-hard problems for very long time, being able to solve them faster has theoretical, as well as practical applications.

[1] For denoting the time we use the notation that suppress polynomial factors.