Efficient algorithms for NP-hard problems on planar graphs

One of the main motivations to study algorithms is to develop efficient procedures to solve intricate optimization problems. Here we would like to provide a computer with a description of, e.g., a road network, and receive a solution to a optimization task which may be, e.g., a shortest path visiting all the cities. For such problems, it is always possible to enumerate all possible solutions and choose the best one but usually the number of solution candidates grows so fast with the growth of the data, that even the most powerful computers turn out hopeless. Therefore, we need not only fast machines but also smart algorithms. Here are two examples of optimization problems related to this project.

Example: capacitated clustering. Imagine that a new city is being built (like in the Cities: Skylines game) and we need to decide where to locate water pumping stations. We estimate the demand to require 10 such stations and in order to optimize the operating costs we would like to minimize the average distance from each receiving point to the corresponding pumping station. Moreover, each station has a bounded capacity, e.g., it could serve at most 100,000 housing units. How to find an optimal or a close-to-optimal construction plan faster than by considering all possible allocations (the number of which is huge)?

Example: circuit routing. Now consider the task of designing a motherboard for a computer. The are several pairs of elements that need to be connected by a circuit, e.g. the processor and the RAM memory slot. Furthermore, we need to place these circuits in such a way that they do not cross or touch each other. How to determine whether this is possible for a given motherboard design?

Common theme: planarity. Even though these two problems look seemingly unrelated, they share a common property. In both cases we need to analyze a graph (that is, a network of connections) that can be drawn in the plane. We call such a graph *planar*. The class of planar graphs enjoys a very interesting mathematical structure, which often comes in useful in the algorithmic design. It has been observed that some computational problems, that are notoriously hard in general graphs, become simpler in planar graphs. Beside the aforementioned examples, such problems occur naturally in transportation, chemistry, image processing, or data visualization.

Scope of the project. The theory of planar graphs has been extensively developed for almost a century, revealing its deep connections to combinatorics, topology, logic, algebra, graph decompositions, and areas of computer science such as preprocessing or algorithms on dynamic data. One theory that is regarded one of the greatest feats of modern mathematics is the Graph Minors project, developed in 1980s and 1990s, which paved the way to apply planarity-based techniques far beyond the class of planar graphs. The two fields of computer science in which studying planarity has been particularly fruitful are approximation algorithms and parameterized complexity, which provide an alternative angle on the complexity of computationally hard problems.

Nevertheless, there remains problems related to planarity for which we still lack a full understanding of their computational complexity. These are fundamental problems involving graph editing, routing, or clustering, which generalize the examples given above. What is more, they include “frontier” problems, for which deciphering their behavior would unlock some barriers for further progress in the theory of algorithms. In this project I aim to deepen our understanding of the combinatorial and topological structure of planar graphs and materialize progress by designing more efficient algorithms for important problems.