Problems and solutions

Many problems in computer science ask how to assign one of many values to each object so that a given list of constraints is satisfied. For example, we may assign school classes to time slots so that classes of each teacher get non-colliding time slots; or assign tasks to computers so that no computer gets two tasks requiring the same resources.

Such problems are studied in an abstract way as *colorings of graphs*: we are given a set of *vertices* (the classes or tasks) and *edges* between them (conflicts of various kinds). We wish to assign *colors* (values like time slots or computer processors) to vertices so that all constraints are satisfied – for example, on each edge, we say the two ends must get different colors.

Spaces of solutions

Such problems often have many solutions, there may be many valid colorings. Some may be closer to others, as when they differ on only a few vertices. We may hence analyze the set of all solutions as a space – for example, asking for paths between solutions. What if we know a current solution, but we want to reach a different one through small steps, like changing only one color at a time? We cannot afford to forget any constraint, so this may be impossible, even if we know both solutions!

Algorithms and topology

For some types of constraints, we know algorithms that can do exactly that – finding paths between known solution – and do that optimally. Surprisingly, even though we ask about finite, discrete sets with finite constraints, the algorithms work by thinking of them much less rigidly. Indeed, we may imagine a graph as a set of totally flexible threads instead of edges. The constraints become obstacles such as tubes – any stretching and bending of threads is allowed, we disallow only crossing walls of the tubes.

It feels like virtually anything can be done with such threads, but clearly we cannot separate a closed thread loop from a tube it goes through, or make it wind twice through the same tube instead – this would require cutting the walls. Topology studies such conditions and their generalizations in higher dimensions. It turns out the very same conditions bind possible paths in spaces of solutions quite tightly.

Our research project

The objective of the project is to better understand how these topological conditions relate to our initial problems. More generally, we want to study the topology of solution spaces and use them to prove useful theorems about the existence of particular solutions, or to design algorithm for solving our problems.

We use computer science to formalize some questions, in particular: for which types of problems and constraints is there an algorithm for finding paths between solutions? Such an algorithm may be directly applicable in some cases, when paths are what we look for, but insight into the spaces of solutions can also allow to search for better ones. Indeed, many algorithms used in practice for solving systems of constraints are based on local search – the idea of improving known solutions step by step.

However, we ask many more mathematical questions, most related to an old, simple conjecture stated by Stephen Hedetniemi in 1966. It says that products of two graphs (which we may think of as conjunctions of two systems of constraints) admit the same colorings as the simpler of the two factors. This surprising question has many deep relations with different mathematical theories, and while we do not hope to resolve it, we believe our methods can make a dent in it, helping us understand the fascinating interplay between combinatorics and topology and providing unpredictable applications on the side.