Exact exponential and parameterized algorithms

Summer Semester 2021

Instructor: László Kozma

Tutorials: Mahmoud Elashmawi

Brief description:

The goal of algorithm design is to come up with efficient algorithms for problems that arise in applications. By efficient, we usually mean "polynomial running time for every instance". Unfortunately several (or even most) of the natural problems that we want to solve are NP-hard, and thus finding polynomial-time algorithms for them is hopeless.

Two broad approaches for dealing with this situation, when approximations are not acceptable, are **exact exponential algorithms**, and **parameterized algorithms**. In the first case, we look for algorithms that solve the problem in exponential time, but faster, and possibly using less space, than a trivial brute force approach. In the second case, we try to identify meaningful parameters, so that the exponential growth in running time is limited to these parameters, thus yielding efficient algorithms in important special cases.

Bibliography:

F. Fomin, D. Kratsch: **Exact Exponential Algorithms**. Springer 2010.

M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh: **Parameterized Algorithms**. Springer 2015.

Topics Covered:

WEEK 1. Course organization. Hard problems. Exponential-time, XP and FPT. Motivating examples. Subset Sum, TSP, Vertex Cover. Branching algorithms. Bounded search trees.

WEEK 2. Branching algorithms. Independent Set. Kernelization. Simple kernel for Vertex Cover. Feedback Vertex Set. Crown decomposition. Linear kernel for Vertex Cover.

WEEK 3. Iterative compression. Vertex Cover. Feedback Vertex Set.

WEEK 4. Measure and conquer. Independent Set. Feedback Vertex Set. Set Cover and Hitting Set.

WEEK 5. Set Cover and Hitting Set branching wrapup. Dynamic programming. Permutation problems. TSP, Feedback Arc Set. Set Cover and Hitting Set (DP).

WEEK 6. Dynamic programming. Graph coloring. Counting perfect matchings in bipartite and general graphs. Ryser formula, inclusion-exclusion.

WEEK 7. Pathwidth. Treewidth. Algorithmic applications.
(Guest lectures by Benjamin Berendsohn.)

WEEK 8. Inclusion exclusion: graph coloring. Feedback vertex set (randomized). Color-coding.

WEEK 9. Color coding vs. divide and conquer. Local search for satisfiability.

WEEK 10. Derandomizing the local search algorithm. Time-space tradeoff.

WEEK 11. Time-space tradeoff: TSP, graph coloring.

WEEK 12. Memorization: Maximum Independent Set. Time-space tradeoff: Subset Sum.
Matrix multiplication-based algorithms.

WEEK 13. Two case studies: 1. graph bandwidth, 2. permutation pattern matching.

WEEK 14. Lower bounds. ETH, SETH, FPT-reductions and W-hierarchy.