Workshop II - Approximation Algorithms Based on Linear Programming

The workshop will be held on Tuesday, September 13th. Register to lecture on September 8-9 upon an announcement from the TA.

1 The Primal-Dual Method

Read Chapter 7 in the book "The Design of Approximation Algorithms" about the Primal-Dual Method and give a brief presentation of the method. Make the explanation "your own" with your own examples, intuition, analogies and visuals.

Remark: The talk will be given by the winner of the explanation challenge competition to be held on Tuesday, September 6. The idea behind the competition is that the audience will benefit from hearing several different perspectives on the method.

2 Approximation Algorithms

Next we describe a couple of optimization problems. The goal is to design efficient approximation algorithms for these problems.

In class on Thursday, September 8, teams of students will attempt to solve the problems. Teams working on the same problem will then discuss their findings with each other. Finally, in the workshop there should be different talks about each problem, addressing some or all of the questions listed next. Students who contributed to the design of the algorithms and their analysis should self-organize and decide who's speaking about what.

Several generic questions:

- Give examples of the problem and find their solutions. How would the problem arise in "real life"?
- Prove that the problem is NP-hard. Does it resemble other NP-hard problems you know?
- Formulate the problem as an integer linear program. Discuss alternative formulations.
- Relax the problem to a real linear program. Discuss alternative relaxations.
- Find the dual program. Can you make sense of it and its connection to the primal program? Does it give you a new perspective on the primal problem?
- Find an efficient rounding algorithm and analyze it and/or find an efficient primal-dual algorithm and analyze it.

2.1 Tree Partitioning Problem

Given as input an undirected graph $G = (V, E)$ with edge costs $w : E \to \mathbb{N}$ and a number $k \geq 1$, the goal is to find a minimum cost collection of vertex-disjoint trees of size k that cover all vertices.

2.2 Survivable Network Design

Given as input a connected undirected graph $G = (\{1, \ldots, n\}, E)$ with edge costs $w : E \to \mathbb{N}$, as well as natural numbers r_{ij} for $1 \leq i < j \leq n$. The goal is to find the lowest cost sub-graph such that there are at least r_{ij} edge disjoint paths between i and j for every two vertices $1 \leq i < j \leq n$.