Course on Approximation Algorithms

Ola Svensson (osven@kth.se)

Autumn 2010

Abstract

The area of approximation algorithms is aimed at giving provable guarantees on the performance of heuristics for hard problems. In this course we will study approximation algorithms for a wide range of problems with the goal to understand the general techniques used when designing and analyzing such algorithms. Below is a list of possible topics.

Combinatorial Algorithms

- **Greedy Algorithms:** First strategy to try when designing algorithms. Applications: *Set Cover*, *Precedence-Constrained Scheduling on Identical Machines.*
- **Approximation To Any Degree:** Some problems allow for arbitrarily good approximation if we are willing to spend more time. Applications: *Knapsack, Scheduling on Identical Machines.*
- **Local Ratio Technique:** Elegant and simple framework to obtaining approximation algorithms for several classes of problems, including packing, covering, and scheduling problems. Applications: *Weighted Vertex Cover, Feedback Vertex Set.*

Linear Programming and Semidefinite Programming Based Algorithms

- **(Randomized) Rounding of Extreme Points:** (Probabilistically) round a fractional solution to an integral one without losing to much. Analyzing the structure of extreme points (solutions) to the LP relaxation often help. Applications: *Vertex Cover*, *Set Cover*, *Scheduling on Unrelated Machines.*
- **Primal-Dual Schema**: Obtain combinatorial algorithms by analyzing the linear program relaxation and its dual. Applications: *Facility Location, Steiner Forests.*
- **Iterative Method/Rounding**: Incrementally build the solution by iteratively solving and rounding linear programs. Applications: *Degree Bounded Spanning Tree, Steiner Network.*
- **Rounding of Semidefinite Programs:** Most successful technique for problems with "soft" constraints, that is constraints that may be violated with some additional cost. Applications: *Max Cut*, *Max 2-Sat.*