Institut for Matematik og Datalogi
Syddansk Universitet

January 27, 2022
JBJ

DM867– Spring 2022 – Weekly Note 1

Welcome to the course DM867 Combinatorial Optimization!

Litterature

- Main text: PS= Papadimitriou and Steiglitz, Combinatorial Optimization – Algorithms and complexity, Dover 2000. Available in the bookstore for only 169 kr. The book is very cheap so even though we will only use a few chapters of it, I strongly recommend that you buy it.
- SCH = Schrijver: A course in Combinatorial Optimization, Lecture Notes CWI 2009. Available for download from the course page.
- KHU = Khuller : Design and Aanalysis of algorithms, course notes University of maryland. Available for download from the course page.
- BJG = Bang-Jensen and Gutin, Digraphs: Theory, Algorithms and Applications, Springer Verlag 2001. Available for download from the course page.
- We will also cover various research papers. These will be made available via the home page when they are needed.

Evaluation
The evaluation is a combination of a "Take home" exam where the students solve one or more sets of problems and an oral exam at the end of the course (as you have tried in DM551 and DM553 and probably other courses). Currently the plan is to have two sets of problems and you may work in groups of up to 3 when solving the first set of problems. The second must be solved alone. The exam is graded after the Danish 7 point scale with external examination.

Format:

The course will be taught physically on campus!

Note, though, that I may decide to do some of the lectures as video lectures, in which case you will also get the slides I use and we will meet physically to discuss the most important parts.

Number of classes
There are more classes scheduled than we need so I will cancel some along the way.

Classes in week 5:

In the first week we start with lectures to give you some new material to work on. Generally I will expect you to have read (most of) the material to be covered at the lecture before showing up. That way we can concentrate more on the difficult parts. You will see from the weekly note what the program for the next week will be.

- I will start by giving an overview of the course and also recapitulate topics such as matchings in graphs and flows in networks. A sample of the topics we will be looking at in the course is:
  - Minimum spanning trees, edge disjoint spanning trees, min-min cost edge-disjoint spanning trees, min-sum edge-disjoint spanning trees.
  - Steiner trees.
  - Flows, min cost flows, shortest paths
  - Matchings and vertex covers
  - Linear programming, duality and primal dual algorithms.
  - Integer Programming and modelling.
  - Connectivity, Menger's theorem, Lovász's splitting theorem, Increasing edge-connectivity optimally, Gomory Hu Trees, Maximum adjacency orderings.
  - Linkages in (di)graphs and Graph Minors.
  - Matroids, greedy algorithm, matroid intersection and union
  - Tree-width and algorithmic applications of tree-width
  - Chordal graphs (graphs with no induced cycle of length 4 or more)
  - Fixed parameter tractability.
  - Various NP-completeness proofs
- Flows in networks. BJG Chapter 3.1-3.5 (mostly 3.1 and 3.5). Even though some (most/all?) of you may know flows already, I will give the necessary background again. In particular I will show how several flow models are equivalent and thus allowing us to use the model which suits the current application best.
- Matroids, spanning trees and the greedy algorithm. PS 12.1-12.4 and SCH 10.1.
- Matchings in bipartite graphs, vertex and edge covers and connectivity of (di)graphs (Menger's theorem). The relevant material is
  - SCH sections 3.1-3.4, 4.1

  - PS 10.1-10.3
  - BJG 3.11.1, 7.3

- To prepare yourself for the first lecture (and the course) you may read the following:

  - PS chapter 1.1-1.2, 2.1-2.2 and 3.1-3.2 (only so that you recall what a linear programming problem and its dual is)
  - SCH pages 5-9
  - Any material you have on minimum spanning trees and shortest paths, so that you are well aware how to solve these problems and remember the algorithms by Dijkstra and Kruskal.