Winter term 2012/13
Juniorprofessor Dr. Stephan Held
Jan Schneider

Research Institute for Discrete Mathematics
University of Bonn

Combinatorial Optimization

Exercise Sheet 9

Exercise 9.1:
Let G be a simple graph with $|V(G)| \geq 2$ and $|\delta(v)| \geq k$ for all $v \in V(G)$. Prove that there are two vertices s and t such that there exist at least k edge-disjoint s-t-paths in G. Is this still true if there is exactly one vertex v with $|\delta(v)| < k$?
Hint: Use Gomory-Hu trees.

(3 Points)

Exercise 9.2:
Show how to solve the MAXIMUM WEIGHT b-MATCHING PROBLEM for the special case where $b(v)$ is even for all vertices v in strongly polynomial time.
Hint: Reduction to the MINIMUM COST FLOW PROBLEM.

(4 Points)

Exercise 9.3:
Let $G = (V, E)$ be a graph. We define $\mathcal{F} := \{X \subseteq V \mid X \text{ is covered by some matching}\}$ and $\mathcal{F}^* := \{X \subseteq V \mid X \text{ is exposed by some maximum matching}\}$

1. Show that $(V, \mathcal{F})$ and $(V, \mathcal{F}^*)$ are matroids. (3 Points)
2. Show that $(V, \mathcal{F}^*)$ is the dual matroid of $(V, \mathcal{F})$. (2 Points)

Exercise 9.4:
Let P be the convex hull of characteristic vectors of independent sets of a matroid $(E, \mathcal{F})$. Prove that $P \cap \{x \in \mathbb{R}^E \mid \sum_{e \in E} x_e = r(E)\}$ is the convex hull of characteristic vectors of bases of a matroid. Devise a polyhedral description of the spanning tree polytope.

(4 Points)

Deadline: Tuesday, December 11, 2012, before the lecture.

Note the programming exercise on page 2!

Programming Exercise 2:

Implement the MINIMUM MEAN CYCLE ALGORITHM from exercise 7.3.

Program Specification: Your program must accept a filename as a command-line parameter (i.e. it must be called with `myprogram input.dmx`). The command-line parameter contains the filename of the file that encodes the graph.

Input: The input file is a DIMACS file that encodes a weighted undirected graph. That means, one line has the format

`p edge` n m

where n is the number of vertices of the graph and m is the number of edges. After this line, m lines have the format

`e` i j c

where i and j are the indices of the vertices connected by this edge and c is the weight of the edge. The vertices are indexed from 1 to n. Lines starting with a `c` are comments and should be ignored. For a more complete definition of the DIMACS format, see `http://www.or.uni-bonn.de/lectures/ss12/praktikum/ccformat.pdf`.

Output: Your program must write the mean weight of a minimum mean cycle to the standard output, followed by a minimum mean cycle (in the DIMACS format).

Perfect Matchings: To compute minimum-weight perfect matchings, you may use existing graph libraries like Lemon (`http://lemon.cs.elte.hu/trac/lemon`) or Blossom V (`http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html`). We provide source code for an example program that reads a DIMACS graph and computes a minimum-weight perfect matching with Blossom V at `http://www.or.uni-bonn.de/lectures/ws12/co_uebung_ws12.html`. Note that Blossom V crashes when the input graph does not have a perfect matching. However, you may also use your own implementation of a MWPM algorithm.

Instances: Test instances are provided together with the example code.

Programming Languages: Your program must be written in C or C++ and compile with a GNU compiler on a current Linux machine.

Criteria: The following criteria are relevant for the number of points you will be awarded: Correctness, speed, code documentation, number of compiler warnings, overall elegance. Note that we use the `-Wall` and `-pedantic` compiler flags.

Submission: Send your program to `schneid@or.uni-bonn.de`.

(20 Points)

Deadline: Tuesday, January 8, 2013, before the lecture.