Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Chris Pinkau, Stefan Toman, Jeremias Weihmann

Sommersemester 2013
Aufgabenblatt 5
13th May 2013

Praktikum Diskrete Optimierung

Due date: Monday, 27th May 2013, 14:00

Aufgabe 1 (Weighted Matchings `weightedmatching`)

Let $G = (V, E)$ be an undirected bipartite graph such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and every edge is adjacent to a node of V_1 and a node of V_2. Let $w(e) \in \mathbb{Z}$ be the edge weight of edge $e \in E$. Notice that negative edge weights are allowed! Implement an efficient algorithm that computes in time $O(|V|^3)$ a maximum matching (i.e. a matching having maximum cardinality) of G which has maximum weight over all maximum matchings of G. (The weight of a matching equals the sum of the weights of its edges.) Extend the algorithm in such a way that the user can choose if a maximum matching with minimum weight or a maximum matching with maximum weight is computed. Your program should output the weight of the computed matching at the end.

Utilize the visualization capabilities of `GraphWin` to vividly visualize how the algorithm works.

Hinweise

You can use the graphs `wbipartite1.gw` to `wbipartite4.gw` as inputs for your algorithm. In this graphs all nodes of V_1 contain the user label "1" while the nodes of V_2 contain the user label "2". The weights of the edges are stored in the user-labels of the edges. All edge weights are numbers between -2 and 17, and your program can visualize these weights well by setting the edge width to the weight of the edge (for edges with negative weights dashed lines can be used.)

The following table shows the weight of a maximum matching with minimal and maximal weight, respectively.

	`wbipartite1.gw`	`wbipartite2.gw`	`wbipartite3.gw`	`wbipartite4.gw`
minimal	29	-2	10	46
maximal	33	-1	22	81