# Re-Exam "Discrete Optimization"

**Friday, February 20, 2015, 13:15 – 16:15**

- Use of calculators, mobile phones, and other electronic devices is not allowed!
- The exam consists of six problems. Please start a new page for every problem.
- The total number of points of the regular assignments is 45. You get 5 bonus points, and you can get 5 bonus points from Exercise 1b.

## 1. Spanning Trees

We consider the problem **ExactST01**:

*Instance:* undirected, connected graph $G = (V, E)$; edge costs $c : E \to \{0, 1\}$, i.e., each edge has costs of either 0 or 1; a number $k \in \mathbb{N}$.

*Goal:* find a spanning tree $T$ of $G$ such that $c(T) = \sum_{e \in T} c(e) = k$ (this means that $T$ contains $k$ edges of cost 1 and $n - k - 1$ edges of cost 0), or conclude that no such tree exists.

As usual, we have $n = |V|$ and $m = |E|$.

(a) *(6 points)* Prove the following statement: For every undirected, connected graph $G = (V, E)$ with edge costs $c : E \to \{0, 1\}$, there exist two numbers $a, b \in \mathbb{N}$ with $a \leq b$ and the following property:

- For every $k \in \{a, a + 1, \ldots, b - 1, b\}$, the graph $G$ contains a spanning tree $T$ with $c(T) = k$.
- The graph $G$ does not contain a spanning tree of weight $k$ for $k < a$ or $k > b$.

(b) *(5 points)* Devise a polynomial-time algorithm for **ExactST01**. Prove that your algorithm is correct and analyze its running-time.

(c) *(5 bonus points)* Devise an algorithm for **ExactST01** that has a running-time of $O(m \log m)$. Prove that your algorithm is correct and satisfies the running-time bound.

If you solve this part correctly, you also get the points of Part (b).

## 2. Traveling Salesman Problem

Our goal is to find an approximation algorithm for **MaxTSP**:

*Instance:* undirected, complete graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}^+$.

*Solution:* a Hamiltonian cycle $H \subseteq E$ of $G$.

*Goal:* **maximize** $w(H) = \sum_{e \in H} w(e)$.

For an instance $G = (V, E)$ and $w$ for MaxTSP, let $H^\star$ be a Hamiltonian cycle of $G$ of maximum weight.

In the following, we assume that the number $n$ of nodes is even.

(a) *(3 points)* Let $M^\star$ be a maximum-weight perfect matching of the graph $G$ with edge weights $w$.

Prove that $w(M^\star) \geq \frac{1}{2} \cdot w(H^\star)$.

(b) *(4 points)* Devise a polynomial-time approximation algorithm for MaxTSP. Your algorithm should output a Hamiltonian cycle $H$ with $w(H) \geq \frac{1}{2} \cdot w(H^\star)$.

## 3. NP-Completeness

The **exact-costs spanning tree problem**, denoted by **ExactST**, is the following decision problem:

*Instance:* undirected graph $G = (V, E)$, edge costs $c : E \to \mathbb{N}$, number $k \in \mathbb{N}$.

*Question:* is there a spanning tree $T$ of $G$ such that $c(T) = \sum_{e \in T} c(e) = k$?

*(7 points)* Prove that ExactST is NP-complete.

*Hint:* You can reduce from **SubsetSum**, which is the following problem and known to be NP-complete:

*Instance:* $n$ items with weights $w_1, \ldots, w_n \in \mathbb{N}$, number $k \in \mathbb{N}$.

*Question:* is there a subset $I \subseteq \{1, \ldots, n\}$ of the items such that $w(I) = \sum_{i \in I} w_i = k$?

*Remark:* Note the difference to Exercise 1 – in Exercise 1, only costs 0 and 1 are allowed; here, arbitrary natural numbers are allowed as costs.

## 4. Minimum-Cost Flows

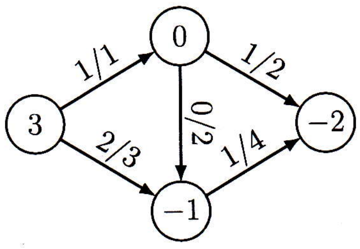
Let $G = (V, E)$ be a flow network with budgets $b : V \to \mathbb{Z}$, capacities $w : E \to \mathbb{N}$, and costs $c : E \to \mathbb{N}$. For a feasible flow $f : E \to \mathbb{R}^+$ on this network, let $H_f = (V, D_f)$ be an undirected graph with the same set $V$ of nodes as $G$ and with the following set $D_f$ of edges:

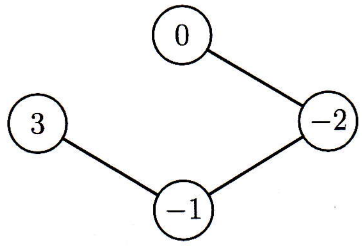
$$D_f = \{\{u, v\} \mid f(u, v) > 0 \text{ and } f(u, v) < w(u, v)\}.$$

This means that the edge set $D_f$ contains an undirected edge $\{u, v\}$ if the edge $(u, v) \in E$ is neither saturated nor empty under the flow $f$.

For simplicity, you can assume that $G$ does not contain pairs of reversed edges, i.e., if $(u, v) \in E$, then $(v, u) \notin E$.

*Example:* The left-hand side shows a flow network including a flow $f$, the right-hand side shows the corresponding graph $D_f$. Budgets are written inside the nodes. The edge labels are "flow/capacity". Costs are omitted as they do not play a role in this example.




(a) *(3 points)* Prove the following: Let $f$ be a feasible flow. If the graph $H_f$ contains a cycle, then $f$ can be written as the convex combination of at least two other feasible flows $g_1$ and $g_2$ such that $|D_{g_1}|, |D_{g_2}| < |D_f|$. This means that the graphs $H_{g_1}$ and $H_{g_2}$ contain fewer edges than $H_f$.

(b) *(3 points)* Prove the following: There exists a feasible flow $f^\star$ of minimum cost such that $D_{f^\star}$ is a forest.

## 5. Matchings

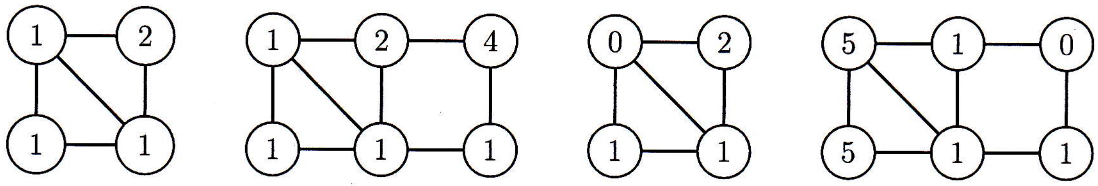
The **node surveillant problem** (denoted by **NodeSurv**) is the following decision problem: An instance of NodeSurv is an undirected graph $G = (V, E)$ with budgets $b_v \in \mathbb{N}$ for all nodes $v \in V$. A node $v \in \mathbb{N}$ can surveil at most $b_v$ of the edges adjacent to it.

Our goal is to assign the edges to the nodes such that as many edges as possible are surveilled.

*(6 points)* Devise a polynomial-time algorithm for determining the maximum number of edges that can be surveilled.

*Remark:* You can use a (polynomial-time) subroutine for computing bipartite matchings of maximum cardinality as a black box. A proof of correctness of your algorithm is not needed, but you should give a brief explanation why it works. A formal analysis of the running-time is also not needed, if it is clear that it is polynomial.

*Example:* In the four examples below, the budgets are written inside the nodes. In the two graphs on the left-hand side, all edges can be surveilled. This means that the answers are "5" and "8". In the third graph, the answer is "4". In the fourth graph, the answer is "7".



## 6. True or False

Are the following statements true or false? Justify your answer.

(a) *(2 points)* Assume that you have a flow network with source $s$ and sink $t$, an $s$-$t$ cut $(X, \overline{X})$, and a flow $f$. Then the capacity of the cut $(X, \overline{X})$ is at least the flow value of $f$, i.e., $c(X, \overline{X}) \geq |f|$.

(b) *(2 points)* If Knapsack can be solved in polynomial time, then 3SAT can be solved in polynomial time.

(c) *(2 points)* Let $G = (V, E)$ be a connected graph consisting of at least three vertices, and let $T$ be a spanning tree of $G$. Then, for all $X \subseteq V$ with $\emptyset \neq X \neq V$, there is exactly one edge in $T$ crossing the cut $(X, \overline{X})$.

(d) *(2 points)* There is a polynomial-time many-one reduction from PerfectMatch $= \{G \mid \text{undirected graph } G \text{ contains a perfect matching}\}$ to 3SAT.