Combinatorische Optimalisering, Exam 25 March 2013

Het tentamen duurt 2 uur en 45 minuten. Studenten die inzage in hun resultaten willen nadat de cijfers bekend zijn kunnen daartoe een afspraak maken per e-mail.

Het tentamen bestaat uit 4 opdrachten opgesplitst in onderdelen. Voor ieder van de in totaal 10 onderdelen kan 10 punten gehaald worden, zoals aangegeven in de volgende tabel:

1	2a	2b	2c	2d	3a	3b	3c	4a	4b
10	10	10	10	10	10	10	10	10	10

Het cijfer wordt verkregen door het behaalde aantal punten door 10 te delen, met een ondergrens van 1. Dus 55 punten zijn nodig om te slagen.

Gedurende het tentamen mag alleen het boek Comb. Opt. by Pap. & Steigl. *zonder losse blaadjes erin op tafel liggen en alle electronische apparaten moeten uitgeschakeld worden.*

---

1. Decide for each of the following statements if it is "true" or "false" without giving any explanation. A correct answer gives 1 point, a mistake gives -1 point. No answer gives 0 points. In the statements a graph refers always to a simple graph, thus not to a multi-graph in which parallel edges are allowed.

(a) There exists a regular graph on k vertices with every vertex having degree d for some odd numbers $k \geq 1$ and $d \geq 1$.
Hint: Remember that a regular graph is a graph with vertices that have all the same degree.

(b) The line graph of a regular graph is regular.

(c) Every complete bipartite graph is regular.

(d) Between each pair of vertices of a tree there exists exactly one path.

(e) Let G be a graph on $n \geq 3$ vertices, each of which has degree at least $\frac{1}{2}n - 1$, then G contains a Hamilton circuit.

(f) A 7-regular graph on 20 vertices contains 70 edges.

(g) Every tree on at least two vertices has at least two points of degree 1.

(h) There exist a disconnected graph on 6 vertices with 11 edges.

(i) Every graph on at least two vertices has two vertices of the same degree.

(j) Every graph on n vertices with m edges has a vertex with degree at most m/n.

Answer. (a)F, (b)T, (c)F, (d)T, (e)F, (f)T, (g)T, (h)F, (i)T, (j)F

2a. Given a complete bipartite graph on two times 5 vertices. The weights of the edges are given in the following table.

	v_1	v_2	v_3	v_4	v_5
u_1	1	3	2	4	5
u_2	4	1	3	2	5
u_3	1	3	2	5	4
u_4	5	2	1	4	3
u_5	4	5	1	3	2

Use the Hungarian Method to find the minimum weight perfect matching in this graph. Show the steps that lead you to the optimal solution.

Answer. Call $u_1, \ldots, u_5$ the nodes corresponding to the rows and $v_1, \ldots, v_5$ the nodes corresponding to the columns. Attach α_i to node u_i and β_j to node v_j. In the initial dual solution choose $\alpha_i = 0\ \forall i$ and set $\beta_j = \min_i c_{ij}\ \forall j$, i.e., $\beta_1 = \beta_2 = \beta_3 = 1$ and $\beta_4 = \beta_5 = 2$. Available edges become $\{u_1, v_1\}$, $\{u_3, v_1\}$, $\{u_2, v_2\}$, $\{u_4, v_3\}$, $\{u_5, v_3\}$, $\{u_2, v_4\}$ and $\{u_5, v_5\}$. Finding a maximum matching on these edges yields e.g. $M = \{\{u_1, v_1\}, \{u_2, v_2\}, \{u_4, v_3\}, \{u_5, v_5\}\}$. Searching an augmenting path from the only exposed u-node u_3 gives a search tree consisting of u_3, v_1, u_1 only. We raise α_1 and α_3 and increase all β_j, $j = 2, 3, 4, 5$ and decrease α_2, α_4 and α_5 and decrease β_1. Determine

$$\theta = \frac{1}{2} \min_{i=1,3,\ j=2,3,4,5} c_{ij} - \alpha_i - \beta_j = \frac{1}{2}$$

The new dual solution is

$$\alpha_1 = \alpha_3 = \frac{1}{2},\ \alpha_2 = \alpha_4 = \alpha_5 = -\frac{1}{2};$$

$$\beta_1 = \frac{1}{2},\ \beta_2 = \beta_3 = 1\frac{1}{2},\ \beta_4 = \beta_5 = 2\frac{1}{2}.$$

The edges $\{u_1, v_3\}$ and $\{u_3, v_3\}$ become available. The search tree starting in u_3 is extended with v_3 and jumping through the matched edge to u_4, where the search stops. Thus we increase α_1, α_3, α_4 and β_2, β_4 and β_5 and decrease α_2 and α_5 and β_1 and β_3. Determine

$$\theta = \frac{1}{2} \min_{i=1,3,4\ j=2,4,5} c_{ij} - \alpha_i - \beta_j = \frac{1}{2}$$

The new dual solution is

$$\alpha_1 = \alpha_3 = 1,\ \alpha_4 = 0,\ \alpha_2 = \alpha_5 = -1;$$

$$\beta_1 = 0,\ \beta_2 = 2,\ \beta_3 = 1,\ \beta_4 = \beta_5 = 3.$$

Among others the edge $\{u_1, v_4\}$ becomes available and we have an augmenting path. Augmentation leads to the perfect matching $M = \{\{u_1, v_4\}, \{u_2, v_2\}, \{u_3, v_1\}, , \{u_4, v_3\}, \{u_5, v_5\}\}$ with total cost 9. Since we have feasible dual and primal solutions that satisfy the complementary slack relations they are optimal.

2b.

Theorem 1. The size of a maximum cardinality matching in a bipartite graph $G = (U, V, E)$ is equal to the size of a minimum cardinality vertex cover.

Prove this theorem.

Answer. See the workout of Exercise 10.1.

2c. Consider finding a maximum weight matching of a bipartite graph, for which all weights of the edges are positive integers. Formulate the dual of this problem and give an interpretation of the dual problem as a type of vertex cover problem. *Hint: For formulating the dual you already gain points.*

Answer. The primal problem is

$$\begin{array}{lll} \max & \sum_{e\in E} w_e x_e & \\ \text{s.t.} & \sum_{e\ni u}^{n} x_e \leq 1, & \forall u \in U; \\ & \sum_{e\ni v}^{n} x_e \leq 1, & \forall v \in V; \\ & x_e \geq 0 & \forall e \in E. \end{array} \tag{1}$$

We know that the constraint matrix, which is the incidence matrix of a bipartite graph, is totally unimodular, hence there exists an optimal solution of the LP that is integral: 0-1.

The dual is

$$\begin{array}{lll} \min & \sum_{u\in U} y_u + \sum_{u\in U} y_u; & \\ \text{s.t.} & y_u + y_v \geq w_e, & \forall e = \{u, v\} \in E; \\ & y_u \geq 0 & \forall u \in U; \\ & y_v \geq 0 & \forall v \in V. \end{array} \tag{2}$$

A feasible solution can be interpreted as numbers on the vertices that together cover the weights of the edges of the graph.

2d. Given the setting of **2c**, use Total Unimodularity to prove that primal and dual problem have integer optimal solutions whose values coincide.

Answer. (Essentially the answer is the same as the second part of the workout of Exercise 10.1.) Since. trivially, the transpose of a totally unimodular matrix is totally unimodular also the above dual problem has an integer optimal solution. Strong LP-duality shows that the values of primal and dual optimal solutions must coincide.

3. Consider the following scheduling problem with time windows.

TIME-WINDOW SCHEDULING:
Instance: Given m parallel identical machines and a set of n jobs and for each job j a processing time p_j, a release date r_j, i.e. the earliest time at which processing

of job j can start, and a deadline d_j, i.e. the latest time by which job j should be completed, $j = 1, \ldots, n$. A feasible schedule of the jobs is an assignment of the jobs to the machines and an ordering of the jobs, such that each job is processed uninterruptedly for a duration of p_j between time r_j and d_j on only one machine and each machine processes at most one job at a time.
Question: Does there exist a feasible schedule?

(**a**) Prove that Time-Window Scheduling is NP-Complete if $m = 2$ (on 2 machines).
Hint: Use a reduction from Partition *or from* Makespan-Decision *(see description of the optimisation version below). A correct set-up of the proof without the exact proof also earns you some points.*

(**b**) Use the result of (**a**) to prove that Time-Window Scheduling on 3 machines is NP-Complete.
Hint: Easiest is to introduce an artificial (long) job in the reduction. Also here, a correct set-up of the proof without the exact proof also earns you some points.

(**c**) Decide for each of the following statements if it is "true" or "false" without giving any explanation. A correct answer gives 1 point, a mistake gives -1 point. No answer is 0 points.

(i) If $\Pi \in P$, then $\Pi \in NP$.

(ii) If $\Pi_1 \in P$ and for every $\Pi_2 \in NP$, $\Pi_2 \propto \Pi_1$, then $P = NP$.

(iii) If Π_1 is NP-complete and $\Pi_1 \propto \Pi_2$, then Π_2 is NP-complete.

(iv) If $\Pi_1 \in NP$ and for every $\Pi_2 \in NP$, $\Pi_2 \propto \Pi_1$, then Π_1 is NP-complete.

(v) If $\Pi_1 \in NP$, Π_2 is NP-complete, $\Pi_1 \propto \Pi_2$, then Π_1 is NP-complete.

Makespan:
Instance: Given a set of n jobs and for each job j a processing time p_j, $j = 1, \ldots, n$ and m parallel identical machines. A feasible schedule of the jobs is an assignment of the jobs to the machines and an ordering of the jobs, such that each job is processed on only one machine and each machine processes at most one job at a time.
Objective: Find a feasible schedule with minimum makespan, i.e. the time the last job completes.

Answer a. First we prove Time-Window Scheduling∈NP: A certificate of a yes-answer is a schedule of the jobs on the machines. It is just a matter of checking if each job is scheduled within its time window, which can be done in $O(n)$ time.

For the reduction define Makespan-Decision as Makespan with an extra parameter K and the question if a schedule exists with makespan at most K. This problem is known to be NP-complete on 2 machines. We will make a reduction from Makespan-Decision on 2 machines to Time-Window Scheduling on two machines.

Take any instance of Makespan-Decision and set for each job j a release time $r_j = 0$ and $d_j = K$. Keep the processing time p_j as it is. Obviously, if the instance of Makespan-Decision has a schedule with makespan at most K then this schedule remains within the time windows set for the instance of Time-Window Scheduling. Reversely, if there is a schedule such that every job of the instance of Time-Window Scheduling remains within its time window then every job completes before time K, hence is a schedule with makespan at most K. Hence yes-answers coincide. Clearly the transformation is executed in $O(n)$ time.

Answer b. That Time-Window Scheduling on 3 machines is in NP follows from **a**.
To make the reduction from Time-Window Scheduling on 2 machines take any instance on n jobs and introduce one extra job with release time $r = \min_{j=1,\ldots,n} r_j$ and deadline $d = \max_{j=1,\ldots,n} d_j$ and give it processing time $p_j = d - r$. This implies that the only way this job can be included in a feasible schedule is by giving it its own machine for its entire processing time. Since its time window $[r, d]$ covers the time windows of all other jobs, the other jobs have to be scheduled feasibly on the two remaining machines. Hence yes-answers coincide. The transformation is clearly computable in $O(n)$ time.

Answer c. (i) true, (ii) true, (iii) false, (iv) true, (v) false

4. Consider the Set Cover problem.

Set Cover:
Instance: Given a universe U of m elements and a collection of sets $S_1, \ldots, S_n$. A *set cover* is a selection of the sets such that each element of U is contained in at least one of the sets in the selection.
Objective: Find a setcover of minimal size, i.e., with a minimum number of sets.

(a) Give the ILP-formulation of this problem, where you use a decision variable x_j for each set S_j, $j = 1, \ldots, n$ and create a restriction for each element of U.

(b) Assume that each element appears in at most k sets. Use rounding of the optimal solution of the LP-relaxation to obtain an approximation algorithm with approximation ratio k.
Hint: It may help you to first consider the case $k = 2$ and then extend it to general k.

Answer a.

$$\begin{aligned} \min \quad & \textstyle\sum_{j=1}^{n} x_j; \\ \text{s.t.} \quad & \textstyle\sum_{j \ni e} x_j \geq 1 \quad \forall e \in U; \\ & x_j \in \{0,1\} \qquad \forall j = 1, \ldots n \end{aligned} \tag{3}$$

Answer b. Suppose that the optimal solution of the LP-relaxation is $\hat{x}$. We round every $\hat{x}_j$ to $\bar{x} = 1$ if and only if $\hat{x} \geq \frac{1}{k}$. Clearly $\sum_{j=1}^{n} \bar{x}_j \leq \sum_{j=1}^{n} k\hat{x}_j = kZ^{LP}$, the optimal value of the LP-relaxation. Since Z^{LP} is a lower bound on the optimal integer solution value, the proof follows if we show that $\bar{x}$ gives indeed a feasible integer solution.

Take any $e \in U$ and suppose without loss of generality that the sets in which it is contained are $S_1, \ldots, S_\ell$ with $\ell \leq k$. Then we have

$$\hat{x}_1 + \hat{x}_2 + \ldots + \hat{x}_\ell \geq 1.$$

In particular this means that at least one of $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_\ell$ must have value at least $\frac{1}{\ell} \geq \frac{1}{k}$. Hence at least one of $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_\ell$ will have value 1, covering element e.