Uitwerkingen van Combinatorische Optimalisering, Tentamen 29 Maart 2012

Het tentamen bestaat uit 4 opdrachten. Het maximum aantal punten dat verdient kan worden voor de verschillende onderdelen staat in de volgende tabel:

1	2a	2b	2c	2d	2e	3a	3b	3c	4a	4b
10	10	10	10	10	5	10	5	10	10	10

Het totale maximum aantal punten 100. Het cijfer wordt verkregen door dit aantal door 10 te delen, met een ondergrens van 1. Dus 55 punten zijn nodig om te slagen.

---

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) Every complete bipartite graph is regular.
Hint: Remember that a regular graph is a graph with vertices that have all the same degree.

(b) A graph is regular if and only if its complement is regular.

(c) A 5-regular graph on 16 vertices has 80 edges.

(d) If in a graph there exists a walk from vertex s to vertex t then there exists a path from s to t.

(e) A graph of which each vertex has degree at least $k-1$, has a path of length k.

(f) There exists a non-connected graph with 6 vertices and 11 edges.

(g) A spanning forest of a graph on n vertices consisting of k components has $n-k-1$ edges.

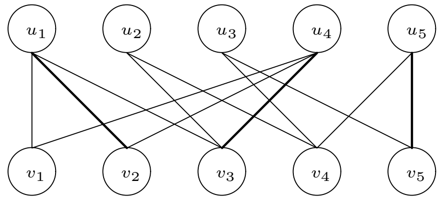
(h) There exist 20 different Hamilton circuits in K_5 (the complete graph on 5 vertices).

(i) There exists a graph with 5 vertices, 3 of which have degree 3.

(j) If a graph has n vertices and m edges and $n > m$ then the graph has at least $n-m$ components.

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

2a. Given is a bipartite graph $G = (U, V)$. The U-vertices are numbered $u_1, u_2, \ldots, u_5$ and the V-vertices are numbered $v_1, v_2, \ldots, v_5$. The boldface printed edges $\{u_1, v_2\}$, $\{u_4, v_3\}$,$\{u_5, v_5\}$ form a matching of cardinality 3 (sorry for not having been able to make them clearer boldface in the picture). Apply the method of augmenting paths to find a perfect matching in G. Show clearly how you find augmenting paths.



Answer. By Breadth First Search find an augmenting path starting from exposed nodes u_2 and u_3 (or v_1 and v_4). In the first step already the augmenting path from u_2 to v_4 is found. Augment the matching by simply inserting the edge $\{u_2, v_4\}$.

Start another augmenting path search, now from the only left over exposed node u_3 (or v_1). This gives first v_4, jumping immediately through the matching edge to u_2, and v_5 jumping immediately through the matching edge to u_5. From u_2 we find v_3, jumping immediately through the matching edge to u_4, whereas the search from u_5 stops, being successful. From u_4 we find v_1 and v_2, the former one being an exposed node. Hence we have found the augmenting path $P = u_3, v_4, u_2, v_3, u_4, v_1$ and we augment the matching by deleting the edges of the matching on P, which are $\{u_2, v_4\}$ and $\{u_4, v_3\}$, and inserting the edges on P not in M, which are, $\{u_3, v_4\}$, $\{u_2, v_3\}$ and $\{u_4, v_1\}$. This is a perfect matching and therefore cannot be augmented any further.

2b. Given a complete bipartite graph on two times 3 vertices. The weights of the edges are given in the following table.

	v_1	v_2	v_3
u_1	9	8	2
u_2	7	6	0
u_3	6	3	1

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. Attach α_i to node u_i and β_j to node v_j. In the initial dual solution choose $\alpha_1 = \alpha_2 = \alpha_3 = 0$ and set $\beta_1 = \min\{c_{11}, c_{21}, c_{31}\} = 6$, $\beta_2 = \min\{c_{12}, c_{22}, c_{32}\} = 3$ and $\beta_3 = \min\{c_{13}, c_{23}, c_{33}\} = 0$. Available edges become $\{u_3, v_1\}$, $\{u_3, v_2\}$ and $\{u_2, v_3\}$. Finding a maximum matching on these edges yields e.g. $M = \{\{u_3, v_2\}, \{u_2, v_3\}\}$. Searching an augmenting path from the only exposed u-node u_1 gives a search tree consisting of u_1 only. We raise α_1 and increase all β_j, $j = 1, 2, 3$ and decrease α_2 and α_3. Determine

$$\theta = \frac{1}{2} \min_{j=1,2,3} c_{1j} - \alpha_1 - \beta_j = \frac{1}{2}(c_{13} - \alpha_1 - \beta_3) = 1.$$

The new dual solution is

$$\alpha_1 = 1,\ \alpha_2 = -1,\ \alpha_3 = -1;$$

$$\beta_1 = 7,\ \beta_2 = 4,\ \beta_3 = 1.$$

The edge $\{u_1, v_3\}$ becomes available. The search tree starting in u_1 is extended with v_3 and jumping through the matched edge to u_2, where the search stops. Thus we increase α_1 and α_2 and β_1 and β_2 and decrease α_3 and β_3. Determine

$$\theta = \frac{1}{2} \min_{1=1,2,\ j=1,2} c_{ij} - \alpha_i - \beta_j = \frac{1}{2}(c_{11} - \alpha_1 - \beta_1) = \frac{1}{2}(c_{21} - \alpha_2 - \beta_1) = 0.5.$$

The new dual solution is

$$\alpha_1 = 1.5,\ \alpha_2 = -0.5,\ \alpha_3 = -1.5;$$

$$\beta_1 = 7.5,\ \beta_2 = 4.5,\ \beta_3 = 0.5.$$

The edges $\{u_1, v_1\}$ and $\{u_2, v_1\}$ become available. Since v_1 is an exposed node, we find the augmenting path immediately and find the perfect matching $M = \{\{u_1, v_1\}, \{u_3, v_2\}, \{u_2, v_3\}\}$ with total cost 12. Since we have feasible dual and primal solutions that satisfy the complementary slack relations they are optimal.

2c. Given a graph $G = (V, E)$ and a subset of the vertices $V' \subset V$, we define $\Gamma(V')$ as the set of all vertices not in V' that are adjacent to some vertex in V':

$$\Gamma(V') = \{v \in V \setminus V' \mid \exists u \in V' \text{ with } \{u, v\} \in E\}.$$

Let $|X|$ denote the cardinality of X, i.e. the number of elements in the set X.

Theorem 1. A bipartite graph $G = (U, V, E)$ has a perfect matching if and only if for every subset $U' \subset U$, we have $|\Gamma(U')| \geqslant |U'|$.

Prove this theorem.

Answer. $\Rightarrow$: This is trivial. If it has a perfect matching then there cannot exist a subset $U' \subset U$ with $|\Gamma(U')| < |U'|$.

$\Leftarrow$: Proof by contradiction. Let $\Gamma(S) = \{v \in V \mid \exists u \in S : \{u, v\} \in E\}$. Thus $|\Gamma(S) \geqslant |S|\ \forall S \subset U$. Suppose that given some maximum cardinality matching M there are exposed vertices $U_0 \subset U$ and $V_0 \subset V$. Doing a breadth first search on alternating paths starting in the exposed U_0-wertices fails to end in any exposed vertex of V, otherwise an augmenting path would have been found. Let U_T be all the vertices of $U \setminus U_0$ on the search tree. Let $V_T = \Gamma(U_T \cup U_0)$ be all V-vertices on the search tree. Thus, $V_0 \cap V_T = \emptyset$. Also, clearly $|V_T| = |U_T|$. Hence, $|\Gamma(U_T \cup U_0)| = |V_T| = |U_T| < |U_T \cup U_0|$, a contradiction. □

2d. Theorem 2. Every k-regular bipartite graph has a perfect matching. *Hint: for the proof you are allowed to use Theorem 1 even if you have not proved that correctly. Of course you are not allowed to use Theorem 3 below ;-)*

Prove this theorem.

Answer. We prove this theorem by proving that every k-regular bipartite graph satisfies the conditions of Theorem 1. Take any set $S \subset V$. Since the graph is k-regular the number of edges from S to $\Gamma(S)$ is $k|S|$. This implies that $\Gamma(S)$ must have at least $k|S|/k = |S|$ vertices. Thus $|\Gamma(S)| \geqslant |S|$. □

2e. Theorem 3. Every k-regular bipartite graph has k edge-disjoint perfect matchings, i.e. k perfect matchings such that none of them has edges that also belong to another one. *Hint: for the proof you are allowed to use Theorem 2 even if you have not proved that correctly.*

Prove this theorem.

Answer. We prove the theorem by induction. The theorem is clearly true for any 1-regular bipartite graph. Now take any k-regular bipartite graph. By Theorem 2 the graph has a perfect matching. Take such a perfect matching and remove all its edges from the graph. This decreases the degree of every vertex by 1.Hence, we obtain a $k-1$-regular graph, which by induction contains $k-1$ edge disjoint perfect matchings. Together with the perfect matching removed they make k edge disjoint perfect matchings. □

3. Consider the MAKESPAN scheduling problem.

MAKESPAN:
Instance: Given a set of n jobs and for each job j a processing time p_j, $j = 1, \dots, n$ 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.

(**a**) Formulate MAKESPAN-DECISION, the decision version of MAKESPAN and show that it is NP-Complete if $m = 2$ (on 2 machines).
Hint: Use a reduction from PARTITION. *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 MAKESPAN-DECISION 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**) Design a PTAS for the MAKESPAN problem on 3 machines. Include the running time analysis.
Hint: The way the PTAS is constructed for the 2-machine version is useful here.

Answer (a). For the decision problem a constant K is introduced and the question is if a schedule of the jobs on the two machines exists such that the makespan is at most K.

MAKESPAN-DECISION∈NP since a Yes-answer has a certificate consisting of a partition of the jobs over the two machines, and one just needs to check if each job occurs once and sum the processing times of the jobs on the two machines separately, select the largest of the two sums and compare to K. Al this takes $O(n)$ time.

PARTITION∝MAKESPAN-DECISION. Take an instance of PARTITION and for each item j with size a_j introduce a job j with processing time $p_j = a_j$. Moreover set K for the instance of MAKESPAN-DECISION equal to $K = \frac{1}{2}\sum_j p_j$. This transformation is done even in constant time.

If the MAKESPAN-DECISION instance has a Yes-answer then this can only be achieved by a partition of the jobs over the 2 machines such that the sums of processing times on the machines are equal, i.e. both equal to $\frac{1}{2}\sum_j p_j$. The corresponding partition of the items gives a Yes-answer for the PARTITION instance.

If the PARTITION instance has a Yes-answer then the partition of the items implies a partition of the corresponding jobs over the two machines such that each machine completes at exactly $\frac{1}{2}\sum_j a_j = \frac{1}{2}\sum_j p_j \leqslant K$. □

Answer (b). That MAKESPAN-DECISIONon 3 machines ∈NP has the same argument than in (a).

MAKESPAN-DECISION on 2 machines $\propto$ MAKESPAN-DECISION on 3 machines. Take an instance of MAKESPAN-DECISION on 2 machines and introduce an auxiliary job with processing time K. Again this is constant time transformation.

The 3rd machine is urged to be used only for the auxiliary job, which shows that Yes-answers between the two instances correspond. □

Answer (c). As a PTAS $A(k)$ we propose to select the k largest jobs, i.e. with largest processing times. Depending on the desired ratio $1+\epsilon$ we will choose k later. We schedule these optimally on the 3 machines. All remaining jobs we schedule according to List Scheduling. The running time of this algorithm is $O(3^k)$ for optimally scheduling the k largest jobs and then $O(n)$ for the others.

We analyze the error made. Suppose that ℓ is the last job to complete. If ℓ belongs to the k largest jobs then we have found the optimal solution. Otherwise,

$$\begin{aligned} Z^{A(k)} = S_\ell + p_\ell &\leqslant \frac{1}{3}\sum_{j=1}^{k} p_j + p_\ell \\ &\leqslant \frac{1}{3}\sum_{j=1}^{n} p_j + \frac{2}{3}p_\ell. \end{aligned}$$

An obvious lower bound for Z^{OPT} is

$$Z^{OPT} \geqslant \frac{1}{m}\sum_j p_j.$$

Hence,

$$\begin{aligned} \frac{Z^{A(k)}}{Z^{OPT}} &\leqslant 1 + \frac{\frac{2}{3}p_\ell}{\frac{1}{3}\sum_{j=1}^{n} p_j} \\ &= 1 + \frac{2p_\ell}{\sum_{j=1}^{n} p_j} \\ &\leqslant 1 + \frac{2p_\ell}{\sum_{j=1}^{k} p_j + p_\ell} \\ &\leqslant 1 + \frac{2p_\ell}{kp_\ell + p_\ell} \\ &\leqslant 1 + \frac{2}{k+1}. \end{aligned}$$

Thus, $A(k)$ has approximation ratio $1 + \frac{2}{k+1}$.

To obtain a ratio of $1+\epsilon$ we need to choose k such that $\epsilon \leqslant \frac{2}{k+1}$ i.e. $k+1 \geqslant \frac{2}{\epsilon}$ or $k \geqslant \frac{2}{\epsilon} - 1$. For ϵ we call the smallest such value $k(\epsilon)$: $k(\epsilon) = \frac{1}{\epsilon} - 1$. Then the algorithm $A(k(\epsilon))$ yields approximation ratio $1+\epsilon$ and its running time is $O(3^{2/\epsilon} + n) = O(3^{1/\epsilon} + n)$. Therefore for all possible values of ϵ the algorithms $A(k(\epsilon))$ form a PTAS.

4. Consider the HAMILTONIAN WALK problem.

HAMILTONIAN WALK:
Instance: Given a (unweighted) graph $G = (V, E)$. A Hamiltonian Cycle in the graph is a closed walk that visits every vertex of the graph *at least* once.
Objective: Find a Hamiltonian Cycle with minimum total length (number of edges/vertices on the closed walk).

(**a**) Prove that this problem is *NP-hard*. *Hint: Easy reduction from Hamiltonian Circuit*

(**b**) Design a polynomial time approximation algorithm A that has an approximation ratio

$$\frac{Z^A(I)}{Z^{OPT}(I)} \leqslant \frac{3}{2}$$

for every instance I of the problem, and prove this ratio. *Hint: Remember metric TSP*

Answer (a). We prove NP-hardness of Hamiltonian Walk by proving NP-completeness of its decision version, which specifies a constant K and asks iif there exists a Hamiltonian Cycle with length at most K.

Hamiltonian Walk-decision∈NP. A yes-certificate is a sequence of vertices (and edges) that claim to be a Hamiltonian Cycle, which can be checked in $O(|H|)$ time, where H is the number of edges on the cycle. We claim that every edge $\{u, v\}$ appears at most twice on the cycle (a proof that it appears at most $2|V|$ times suffices as well). This makes that $|H| \leqslant 2|E|$ completing the proof.

To prove the claim: suppose that a cycle exists that visits edge $\{u.v\}$ more than twice. If, following the cycle in some direction, $\{u, v\}$ is traversed at least once from u to v and at least once from v to u, then there exist a subcycle $v, v_1, \ldots, v_i, v$, which can be traversed the third time the edge $\{u, v\}$ is traversed in whatever direction. This diminishes the number of times that $\{u, v\}$ is traversed. Otherwise there are at least 3 times that the edge $\{u, v\}$ is traversed from u to v. Then there exist subcycles $u, v, v_{11}, \ldots, v_{1i_1}, u$ and $u, v, v_{21}, \ldots, v_{2i_2}, u$ which can be combined into $u, v_{1i_1}, \ldots, v_{11}, v, v_{21}, \ldots, v_{2i_2}, u$ also diminishing the number of times $\{u, v\}$ is traversed.

For NP-completeness we make Hamiltonian Circuit∝Hamiltonian Walk-decision, simply by defining $K = n$ with $n = |V|$. Obviously, there exists a Hamiltonian Circuit in a graph if and only if the graph has a Hamiltonian Cycle of length n. □

Answer (b). The algorithm we propose is like Christofides' algorithm for the metric TSP. Assume that the graph is connected. Otherwise apply the algorithm to the components separately. Thus, first find a spanning tree T of G (notice that every spanning tree is a minimum spanning tree) in $O(|E|)$ time. Then on the nodes of T with odd degree find a minimum perfect matching, with the cost of a connection between two nodes chosen as the shortest path between the two nodes. Add the shortest paths corresponding to the matching to T. Let this matching (the union of the shortest paths) be M. This gives a Eulerian graph, the Eulerian walk of which is a Hamiltonian Cycle.

The length of the Hamiltonian Cycle is clearly $c(T) + c(M) = n - 1 + c(M)$. Clearly, $Z^{OPT} \geqslant n$. Hence we need to argue that $Z^{OPT} \geqslant \frac{1}{2}c(M)$. Take any shortest Hamiltonian cycle on G. Consider it as a multi-graph. Take any of the two orientations and start in an odd vertex v of T. Color the path from v to the next odd vertex v' of T on the cycle red and the path from v' to the next odd vertex v'' of T, not visited before, on the cycle blue. Continue until v has been reached again for the last time. This must be by a blue path, since there are an even number of odd vertices of T. By then all (multiple) edges of the Hamiltonian cycle have been colored red or blue. The number of blue edges and the number of red edges is equal to the length of the shortest Hamiltonian cycle. Both the red paths and the blue paths correspond to perfect matchings of the odd vertices of T. Therefore one of them is such a matching with at most $1/2$ the number of edges as the shortest Hamiltonian Cycle.