Master 2
Graph Algorithms and Combinatorial Optimization
Final Exam, November 2022
3 hour
No documents nor electronic devices allowed.

Instruction and comments: the points awarded for your answer will be based on the correctness of your answer as well as the clarity of the main steps in your reasoning. All proposed solutions must be proved. **All the exercises are independent.** *The points are indicated so you may adapt your effort.*

1 Linear Programing

Exercise 1 (2 points, 15 minutes) Let $D(V, A)$ be a digraph. We distinguish four vertices $s_1, s_2, t_1, t_2 \in V$. Write a linear program finding two node-disjoint paths, the first one between s_1 and t_1 and the second one between s_2 and t_2, if such paths exist (two node-disjoint paths do not share any node). *Explain all the lines of your linear program.*

Exercise 2 (Bin Packing problems. (4 points, 30 minutes)) We consider the following packing problem. We are given n items. Item j has size p_j $(j = 1, ..., n)$. These items have to be divided over m bins. The goal is to assign the items to the bins in such a way that $P_{\max}$ is minimized, where $P_{\max}$ equals the maximum bin filling. The bin filling is defined as the sum of the sizes of the items in one bin.

a) Give an integer linear programming formulation for this problem with a polynomial number of variables.

b) Now we are allowed to buy two additional bins. The first additional bin costs Q_1 and the second costs Q_2, where $Q_2 < Q_1$. The second additional bin can only be used in combination with the first. Extend the model of part (a) with the possibility of extra bins. The goal is to minimize the sum of $P_{\max}$ and the cost of the extra bins.

c) Now we consider a variant of the problem of part (a). Suppose our bins have a maximum capacity of C and we want to minimize the number of bins needed to pack all the items. We can formulate this problem by using sets of items assigned to one bin. Give an integer linear programming formulation for the problem based on sets of items.

d) Describe how the LP-relaxation of this formulation can be solved by column generation. Your description should include a formulation of the pricing problem, and an algorithm to solve it.

Exercise 3 (Energy Constrained Maximum Flow. (4 points, 30 minutes)) We consider the energy constrained max-flow problem. We are given a directed graph (V, A), where V is the set of nodes and A is the set of arcs. There is a source node $s \in V$ and a sink $t \in V$. Each node i has a battery with capacity E_i. Sending flow on edge (i, j) requires energy from the battery at node i, which amounts e_{ij} per unit flow. The objective is to find the maximal flow s to t, satisfying the energy constraints, where the flow is required to be integral.

(a) Give an integer linear programming formulation for this problem with a polynomial number of variables.

(b) A network flow can be decomposed into a number of st-paths. An alternative way to formulate the problem is by using these paths. Give an integer linear programming formulation for the problem of part (a) based on paths.

(c) Describe how the LP-relaxation of this formulation can be solved by column generation. Your description should include a formulation of the pricing problem. Describe how to solve the pricing problem.

Exercise 4 (Maximum Average Degree. (2 points, 15 minutes)) The average degree of a graph G is defined as $ad(G) = \frac{2|E(G)|}{|V(G)|}$. The maximum average degree of G is meant to represent its densest part, and is formally defined as:

$$mad(G) = \max_{H \subseteq G} ad(H).$$

Even though such a formulation does not show it, this quantity can be computed in polynomial time through Linear Programming.

Indeed, we can think of this as a simple flow problem defined on a bipartite graph. Let D be a directed graph whose vertex set we first define as the disjoint union of $E(G)$ and $V(G)$. We add in D an edge between $(e, v) \in E(G) \times V(G)$ if and only if v is one of e's endpoints. Each edge will then have a flow of 2 (through the addition in D of a source and the necessary edges) to distribute among its two endpoints. We then write in our linear program the constraint that each vertex can absorb a flow of at most z (add to D the necessary sink and the edges with capacity z).

1) Show that the flow is feasible if and only if $z \geq \frac{2|E(G)|}{|V(G)|}$.

2) Write the linear program modeling the problem.