Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Operations Research
Department of Mathematics

Introduction to Mathematical Optimization

Final Exam

February 08, 2021

Legi.-No.:	

General remarks.

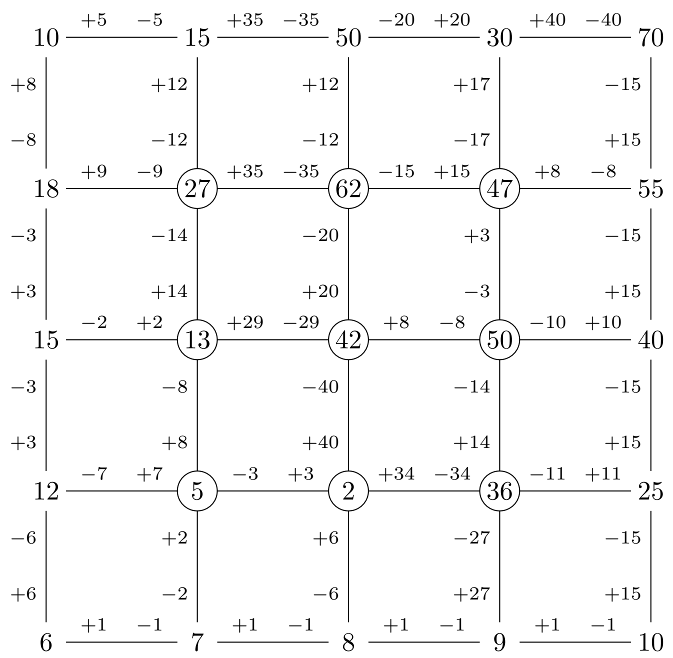
- Place your student identity card on the table.
- Use a pen to write down your solutions. No pencils and no red colored pens are allowed.
- Turn off all your technical devices. Put away everything, except for writing material.
- Write the solutions on the exam. Extra paper will be distributed if needed.
- You have 2 hours to solve the exam.
- For all problems, provide a complete solution in English, including all explanations and implications in a mathematically clear and well-structured way. Please hand in a readable and clean solution. Cross out any invalid solution attempts.
- Auxiliary statements that you use to solve a problem must be proved. Unless you are explicitly asked to prove them, results known from the lectures can be used without a proof. In each such case, state the result or its name.

Good luck!

Problem:	1	2	3	4	5	Total
Points:	/ 10	/ 10	/ 10	/ 10	/ 10	/ 50

Problem 1: Designing Terrains (10 points)

Suppose you are designing terrain for a video game. The terrain is stored as a height map modelled by an $n \times n$ matrix where each cell is assigned a height. In addition to heights of cells, you take into account elevation changes between neighboring cells. More specifically, every cell (i, j) has up to four slopes associated with it, which correspond to (signed) height differences between this cell and the cells adjacent to it horizontally and vertically (i.e., when switching rows and columns). A cell is called internal if it is not on the border of the matrix (i.e. it is not in the first row, first column, last row, or last column). The following picture shows an example of a terrain with $n = 5$. The slopes associated with each cell are written next to the cell and the internal cells are circled.



To create a realistic map suitable for the game, you are tasked with creating a height map that satisfies the following conditions.

- The heights of cells $(i_1, j_1), \ldots, (i_k, j_k)$ are predetermined and should be set to the values $h_1, \ldots, h_k$, respectively.
- The height of every cell should be in the range $[0, 8848]$.
- For every internal cell, the slopes associated with it cannot differ by more than 50 units (in other words, they should lie in an interval of length 50). This condition does not apply to border cells. Note, for example, that the central cell in the figure above violates this constraint, since it has the two slopes $+20$ and -40 associated with it, which differ by more than 50.

Write a linear program that determines whether there exists a height map satisfying the above conditions and, if one exists, computes a map with the smallest possible maximal height.

Problem 2: Cash-Flow Matching (10 points)

Suppose that you decided to invest into bonds and use the returns on these bonds to make certain payments for the next 5 years (for example, taxes). The table below lists the payouts (in CHF) for all of the available bonds and the payments (in CHF) that need to made each year. For example, one unit of bond 1 provides an income of CHF 5 in years 1, 2, and 3, then CHF 55 in year 4, and CHF 0 in year 5.

Year	**1**	**2**	**3**	**4**	**5**
Bond 1	5	5	5	55	0
Bond 2	10	10	60	0	0
Bond 3	5	55	0	0	0
Bond 4	12	12	12	12	12
Payments	50	100	50	150	50

In addition to the above table, you know the following.

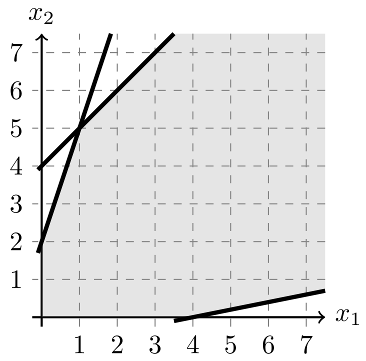
- A single unit of each bond costs CHF 45.
- Bonds can only be bought before year 1 and cannot be sold.
- You have a sufficient amount of money to afford any amount of any bond.
- Bonds 2 and 4 can only be purchased in integer quantities, while bonds 1 and 3 can be purchased in real quantities. For example, you cannot buy 0.5 units of bond 2, but you can buy 0.5 units of bond 3.
- Bonds 3 and 4 are offered by two different banks and you would like to manage your investment through only one of these banks. As a result, you can buy either bond 3, or bond 4, or neither, but you cannot invest into both of them simultaneously.
- You can carry over excess returns on the bonds to the next year. In other words, if your income from bonds for one year exceeds the liabilities for that year, then the remaining money can be used to cover part of next year's payment. The carry over for year 1 is zero.
- Unlike the excess income, the yearly payments cannot be carried over: you have to pay the exact amount stated in the table each year.

Your goal is to create a bond portfolio of minimal cost that covers all the liabilities, i.e., you need to decide how much of each bond to buy before year 1, so that in each year, the total income from bonds for that year plus the carry over amount from the previous year suffices to make that year's payment. Write a mixed-integer linear program to model this problem.

Problem 3: Simplex Method and Duality (10 points)

Consider the LP below on the left with the feasible region depicted in the figure below on the right.

$$
\begin{array}{lrcrcr}
\min & 2x_1 & - & x_2 & & \\
\text{s.t.} & x_1 & - & 5x_2 & \leq & 4 \\
 & 3x_1 & - & x_2 & \geq & -2 \\
 & x_1 & - & x_2 & \geq & -4 \\
 & x_1 & & & \geq & 0 \\
 & & & x_2 & \geq & 0
\end{array}
$$



(a) Rewrite this LP in standard form. (1 pts)

(b) Write down the (long) tableau that corresponds to the origin $(x_1, x_2) = (0, 0)$. Is the tableau feasible? Justify your answer. (1 pts)

(c) Compute an optimal solution of the LP with the Simplex Algorithm. (3 pts)

(d) Write down the dual LP. (1 pts)

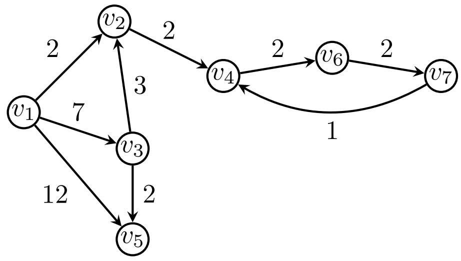
(e) Find an optimal solution of the dual LP using the complementary slackness conditions. (2 pts)

(f) Suppose the objective of the original LP is changed to "max $2x_1 - x_2$". Is the dual of the modified LP feasible? Justify your answer. (2 pts)

Problem 4: Combinatorial Optimization (10 points)

Let $G = (V, A)$ be a directed graph shown in the figure below with arc lengths written next to them.

(a) Run Dijkstra's algorithm to determine all shortest path lengths from v_1 to all the nodes in the graph. For this purpose complete the table given below. After filling in the table, write the distance from v_1 to every other vertex, along with a corresponding shortest path. (5 pts)



current node		v_1	v_2	v_3	v_4	v_5	v_6	v_7
initialization	distance from v_1: visited (yes/no):							
	distance from v_1: visited (yes/no):							
	distance from v_1: visited (yes/no):							
	distance from v_1: visited (yes/no):							
	distance from v_1: visited (yes/no):							
	distance from v_1: visited (yes/no):							
	distance from v_1: visited (yes/no):							
	distance from v_1: visited (yes/no):							

- Distance and a shortest path from v_1 to v_2:
- Distance and a shortest path from v_1 to v_3:
- Distance and a shortest path from v_1 to v_4:
- Distance and a shortest path from v_1 to v_5:
- Distance and a shortest path from v_1 to v_6:
- Distance and a shortest path from v_1 to v_7:

(b) Suppose an arc (v_3, v_4) with negative length -4 is added. For which vertices (if any) does the distance from v_1 decrease after this change? (3 pts)

(c) Starting from the original graph G, if we add the arc $a = (v_7, v_5)$, what are all arc lengths $\ell \in \mathbb{R}$ (positive or negative) that we can assign to a, to ensure that the shortest path length from v_1 to v_5 strictly decreases? (2 pts)

Problem 5: Theory (10 x 1 = 10 points)

For each of the following statements, state if it is **true or false** and provide a short **justification**. No points are given for unjustified, or wrongly justified answers.

1. The set $A := \{(x, y) \in \mathbb{R}^2_{\geq 0} \mid y \leq x^2 + x\}$ is the feasible region of a linear program.

   ☐ **TRUE** ☐ **FALSE**

2. Let $P := \{x \in \mathbb{R}^2 \mid Ax \leq b\}$ and $Q := \{x \in \mathbb{R}^2 \mid Cx \leq d\}$ be two convex polyhedra. Then the set $P \setminus Q := \{x \in \mathbb{R}^2 \mid x \in P \text{ and } x \notin Q\}$ is a polyhedron, i.e., can be written in terms of finitely many linear constraints.

   ☐ **TRUE** ☐ **FALSE**

3. Consider the linear program $\{\max c^T x \mid Ax \leq b\}$ for some $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. Let x_1, x_2, $x_3 \in \mathbb{R}^n$ be distinct points, i.e., $x_1 \neq x_2$, $x_1 \neq x_3$, $x_2 \neq x_3$. It can happen that the set of optimal solutions is exactly $\{x_1, x_2, x_3\}$.

   ☐ **TRUE** ☐ **FALSE**

4. Consider the linear program $\{\max c^T x \mid Ax \leq b\}$ for some $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. Let $x_1,\ x_2,\ x_3 \in \mathbb{R}^n$ be distinct points, i.e., $x_1 \neq x_2,\ x_1 \neq x_3,\ x_2 \neq x_3$. It can happen that the set of optimal solutions is exactly the convex hull of $x_1,\ x_2,\ x_3$, i.e. the set $\{\lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \mid \lambda_1 + \lambda_2 + \lambda_3 = 1,\ 0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1\}$.

☐ **TRUE** ☐ **FALSE**

5. Consider the linear programs $P : \{\max\ c_1 x + c_2 y \mid y - 2x \leq 1,\ 2y + 3x \leq 6,\ x, y \in \mathbb{R}_{\geq 0}\}$ and $P_{\text{int}} : \{\max\ c_1 x + c_2 y \mid y - 2x \leq 1,\ 2y + 3x \leq 6,\ x, y \in \mathbb{Z}_{\geq 0}\}$, for some $c_1, c_2 \in \mathbb{Z}$. Then for every $c_1, c_2 \in \mathbb{Z}$ the optimal value of P is equal to the optimal value of P_{int}.

☐ **TRUE** ☐ **FALSE**

6. Consider the following long tableau.

z	x_1	x_2	x_3	x_4	1
1	-1	0	8	0	0
0	6	9	$\boxed{3}$	0	12
0	2	0	-4	1	1
0	4	-3	2	0	7

Performing a pivot on the circled element leads to a feasible tableau.

☐ **TRUE** ☐ **FALSE**

7. Let $f(n) = 100n^2 \log n$ and $g(n) = n^{2.5}$. Then, $f(n) = O(g(n))$.

☐ **TRUE** ☐ **FALSE**

8. Let G be a graph with $n \geq 1$ vertices and $n - 1$ edges. Then G has no cycles.

☐ **TRUE** ☐ **FALSE**

9. There exists a 5-regular graph with 9 vertices. (Reminder: a k-regular graph is a graph such that each of its vertices has degree k).

☐ **TRUE** ☐ **FALSE**

10. Consider a directed graph $G = (V, A)$ with arc lengths $\ell : A \to \mathbb{Z}_{\geq 0}$. Let v, w, u be three distinct nodes in V. Assume that the shortest v-w path has length p and that the shortest v-u path has length q. Then the shortest w-u path necessarily has length $|p - q|$.

☐ **TRUE** ☐ **FALSE**