

Exercises for Mathematical Programming

186.835 VU 3.0 – SS 2021

Upload your solutions in TUWEL until the following due dates:

- Exercises 1–3: **March 24**
- Exercises 4–5: **April 14**
- Exercises 6–8: **May 5**
- Exercises 9–10: **May 26**
- Exercises 11–12: **June 16**

Notes:

- Cite used references!
 - If you work in a team then submit only one solution and list all participants!
 - Do not forget to define and describe all variables and constraints!
-

Exercise 1 *Network Design (1 point)*

We are given a directed graph $G = (V, A)$ and a value $b_i \in \mathbb{R}$ for each node $i \in V$ which denotes a demand ($b_i < 0$) or a supply ($b_i > 0$), such that $\sum_{i \in V} b_i = 0$. There are two types of costs: transportation costs c_{ij} of shipping one unit from node i to node j , and building costs of establishing a direct link (i, j) from node i to j . A link on arc (i, j) can be built (i) with costs d_{ij}^1 and capacity u_{ij}^1 , or (ii) with costs d_{ij}^2 and capacity u_{ij}^2 . Assume that $d_{ij}^1 < d_{ij}^2$ and $u_{ij}^1 < u_{ij}^2$ and that at most one option can be chosen on each arc. A network has to be built that satisfies all demands and minimizes the total building and transportation costs. Formulate the problem as a (mixed) integer linear program.

Exercise 2 *Scheduling (1 point)*

A factory consists of m machines M_1, \dots, M_m . Each job $j = 1, \dots, n$ needs to be processed on each machine in the specific order $M_{j(1)}, \dots, M_{j(m)}$ (permutation of the machines). Machine M_i takes time p_{ij} to process job j . A machine can only process one job at a time, and once a job is started on any machine, it must be processed to completion. The objective is to minimize the **sum of the completion times** of all the jobs. The completion time of job j is the time when the last subtask on machine $M_{j(m)}$ is finished. Formulate the problem as a (mixed) integer linear program.

Exercise 3 *Sports League (3 points)*

The season in a sports league with n teams has started. Each team plays against each other team at home and away, so each team plays exactly $2(n - 1)$ games. In case of a win a team receives three points, in case of a draw one point, and in case of a loss zero points. After the season the k worst teams with respect to the total number of achieved points get relegated. The task is now to determine the minimum number of points a team has to achieve to have a guarantee that it is not relegated in any season outcome.

- Formulate the problem as a (mixed) integer linear program. **(1 point)**
 - Solve the problem for $n = 18$ and $k = 3$ with an MILP solver (e.g., CPLEX, Gurobi, Excel Solver). **(2 points)**
-

Exercise 4 *Cycle-Elimination Cuts (2 points)*

Consider the polyhedra of the following two ILP formulations for the minimum spanning tree problem:

- cycle elimination formulation (CEC)

$$\min \sum_{e \in E} w_e x_e \quad (1)$$

$$\text{s.t. } \sum_{e \in E} x_e = n - 1 \quad (2)$$

$$\sum_{e \in C} x_e \leq |C| - 1 \quad \forall C \subseteq E, |C| \geq 2, C \text{ forms a cycle} \quad (3)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (4)$$

- subtour elimination formulation (SUB)

$$\min \sum_{e \in E} w_e x_e \quad (5)$$

$$\text{s.t. } \sum_{e \in E} x_e = n - 1 \quad (6)$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \quad \forall S \subseteq V, S \neq \emptyset \quad (7)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (8)$$

Prove or disprove:

$$P_{\text{cec}} = P_{\text{sub}}$$

Exercise 5 *(Prize Collecting) Steiner Tree Problem (2 points)*

Consider the *Steiner tree problem on a graph (STP)* and the *Prize Collecting Steiner tree problem on a graph (PCSTP)*, which are defined as follows:

- STP: Given an undirected graph $G = (V, E)$ with edge weights $w_e \geq 0, \forall e \in E$, whose node set is partitioned into terminal nodes T and potential Steiner nodes S , i.e. $S \cup T = V, S \cap T = \emptyset$. The problem is to find a minimum weight subtree of G that contains all terminal nodes.
- PCSTP: Given an undirected graph $G = (V, E)$ with edge weights $w_e \geq 0, \forall e \in E$, and node profits $p_i \geq 0, \forall i \in V$. The problem is to find a subtree $G' = (V', E')$ of G that yields maximum profit, i.e. $\max \sum_{i \in V'} p_i - \sum_{e \in E'} w_e$.

Provide ILP formulations for each problem using *directed cutset constraints*.

Exercise 6 *Chvátal-Gomory Cutting Planes (1 point)*

Consider the set $X = \{(x_1, x_2) \in \mathbb{Z}^2 : -4x_1 + 6x_2 \leq 9, 2x_1 + 3x_2 \leq 10, x_1 \geq 0, x_2 \geq 0\}$. Use the Chvátal-Gomory procedure to find a description for $\text{conv}(X)$.

Exercise 7 *Cover Inequality (1 point)*

Consider the knapsack set $X = \{\{0, 1\}^6 : 12x_1 + 9x_2 + 7x_3 + 5x_4 + 5x_5 + 3x_6 \leq 14\}$.

- Find a minimum cover C and a corresponding valid cover inequality.
- Find the extended cover inequality for C .
- Lift the inequality to obtain a strong valid inequality, i.e., compute all lifting coefficients.

Exercise 8 *Totally Unimodularity (2 points)*

Prove or disprove

- The node-edge incidence matrix of all simple, undirected graphs is totally unimodular. **(1 point)**
(The node-edge incidence matrix has one column for each edge $\{i, j\}$ with entries $+1$ in rows i and j . A graph is called simple, if it does not contain self-loops or multiple edges between two nodes.)
- The node-arc incidence matrix of all bipartite, directed, simple, graphs is totally unimodular. **(1 point)**
(The node-arc incidence matrix has one column for each arc (i, j) with entries $+1$ in row i and -1 in row j . A directed graph is called simple, if it does not contain self-loops or multiple arcs with the same orientation between two nodes.)

Exercise 9 *Lagrangian Relaxation for Uncapacitated Facility Location (1 point)*

Consider the following formulation for the *Uncapacitated Facility Location Problem* (see 03_lagrangian_relaxation.pdf, p.7-10, for details):

$$\max \sum_{i \in M} \sum_{j \in N} p_{ij} x_{ij} - \sum_{j \in N} f_j y_j \quad (9)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in M \quad (10)$$

$$x_{ij} \leq y_j \quad \forall i \in M, j \in N \quad (11)$$

$$x_{ij} \geq 0 \quad \forall i \in M, j \in N \quad (12)$$

$$y_j \in \{0, 1\} \quad \forall j \in N \quad (13)$$

Your task is now to dualize the linking constraints $x_{ij} \leq y_j$, $\forall i \in M, \forall j \in N$, in the usual Lagrangian way, and answer the following questions:

- From a theoretical point of view, how strong is the best Lagrangian dual bound achievable by this relaxation compared with the optimal LP relaxation value of the original formulation? Can it get weaker, is it always equal, or can it get stronger? Argue your answer!
- How easy is it to solve the Lagrangian subproblem(s) in terms of computational complexity? No proofs required, just sketch the idea!

Exercise 10 *Lagrangian Relaxation for Knapsack Problems (2 points)*

Consider the 0-1 knapsack problem

$$\max 10y_1 + 4y_2 + 14y_3 \quad (14)$$

$$3y_1 + y_2 + 4y_3 \leq 4 \quad (15)$$

$$y_1, y_2, y_3 \in \{0, 1\}. \quad (16)$$

Dualize the knapsack constraint (15) in the usual Lagrangian way.

- What is the optimal value of the Lagrange multiplier u and the value of the Lagrangian dual w_{LD} ?
- Run the subgradient algorithm using the step size method according to rule (2) (see 03_lagrangian_relaxation.pdf, p.23) with $u^0 = 0$, $\mu_0 = 1$, $\rho = 0.5$. Does the subgradient algorithm converge to the best value w_{LD} ? Are the optimal solutions of the Lagrangian subproblems unique?

Exercise 11 Bin Packing (2 points)

In the bin packing problem, we are given a set of bins of capacity W and a list of n items of integer sizes $L = \{l_1, \dots, l_n\}$, $0 \leq l_j \leq W$ for $1 \leq j \leq n$. The objective is to assign the items to the bins so that the capacity of the bins is not exceeded and the number of used bins is minimized.

One classic formulation for the bin packing problem is given by (17)–(21). In (17)–(21), variables $y_i \in \{0, 1\}$, $i = 1, \dots, n$, indicate whether or not bin i is used, while variables $x_{ij} \in \{0, 1\}$, $i, j = 1, \dots, n$, indicate whether or not item j is assigned to bin i .

$$\min \quad z = \sum_{i=1}^n y_i \tag{17}$$

$$\text{s.t.} \quad \sum_{j=1}^n l_j x_{ij} \leq W y_i \quad i = 1, \dots, n \tag{18}$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \tag{19}$$

$$y_i \in \{0, 1\} \quad i = 1, \dots, n \tag{20}$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n \tag{21}$$

- Develop a formulation for the bin packing problem that utilizes an exponential (in the number of items) number of variables.
- Formally state the pricing subproblem that arises when solving the LP relaxation of your formulation by column generation, as well as a feasible approach for solving it.
- Find an instance for which your extended formulation yields a better LP bound than formulation (17)–(21). (If no such example exists, your extended formulation may be wrong / not reasonable.)
- Describe a reasonable branching rule that may be used in a branch-and-price algorithm, as well as its (potential) impact on the pricing subproblem.

Exercise 12 Hop-constrained STP (2 points)

Consider the Hop-constrained Steiner Tree Problem (HCSTP): We are given an undirected graph $G = (V, E)$, a dedicated root node $0 \in V$, a set of terminal nodes $T \subset V \setminus \{0\}$, an edge costs $c_e \geq 0$, $\forall e \in E$, and a hop limit $H \in \mathbb{N}$. A feasible solution S to the HCSTP is a Steiner tree containing the root node and all terminal nodes (further potential Steiner nodes may be included) which satisfies the hop constraint, i.e., the unique path from 0 to any terminal $t \in T$ in S may not contain more than H edges. The objective is to identify a feasible solution $S^* = (V^*, E^*)$ yielding minimum total costs, i.e., $\sum_{e \in E^*} c_e$.

- Provide a *directed* ILP formulation for the HCSTP using an exponential number of *path variables*. (Don't forget to define and describe the variables that you use).
- Formally state the pricing subproblem. (You first need to describe which dual variables are associated to which of your constraints.)
- Give an interpretation of the pricing subproblem. (Which combinatorial optimization problem needs to be solved?). Can we solve pricing subproblem in polynomial time or is it NP-hard? (Hint: You may need to argue on the sign of a subset of your dual variable values.)