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^1_{ij} and capacity u^1_{ij}, or (ii) with costs d^2_{ij} and capacity u^2_{ij}. Assume that $d^1_{ij} < d^2_{ij}$ and $u^1_{ij} < u^2_{ij}$ 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. **Answer:** Assume we model a arc of type 1 between (i, j) by the variable $x^1_{ij} \in \{0, 1\}$ and of type 2 respectively by $x^2_{i,j} \in \{0, 1\}$. Then the objective function is given by :

$$g(x^1, x^2) = \sum_i \sum_j \left[x^1_{ij}(c_{ij}u^1_{ij} + d^1_{ij}) + x^2_{ij}(c_{ij}u^2_{ij} + d^2_{ij}) \right] \tag{1}$$

Therefore we get the following MILP:

$$\min_{x_1, x_2} g(x^1, x^2) \tag{2}$$

$$\sum_j (x^1_{ij}u^1_{ij} + x^2_{ij}u^2_{ij}) \le b_i \quad \forall i \tag{3}$$

$$\sum_j (x^1_{ij}u^1_{ij} + x^2_{ij}u^2_{ij}) \ge b_i \quad \forall i \tag{4}$$

$$x^1_{ij} \ge 0 \quad \forall i, j \tag{5}$$

$$x^2_{ij} \ge 0 \quad \forall i, j \tag{6}$$

$$x^1_{ij} + x^2_{ij} \le 1 \quad \forall i, j \tag{7}$$

In (3) and (4) we ensure that the demand in every node is satisfied. The constrant (7) only allows either a type 1 or type two connection is build.

Exercise 2 *Scheduling (**1 point**)*

A factory consists of m machines $M_1, \ldots, M_m$. Each job $j = 1, \ldots, n$ needs to be processed on each machine in the specific order $M_{j(1)}, \ldots, M_{j(m)}$ (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. Assume the variable x_{kj} defines the start time of a job j on machine k. Jobs cant overlap on a machine, and we want that the condition $(x_{kj} - x_{ki} \geq p_{ki}) \vee (x_{ki} - x_{kj} \geq p_{kj})$ holds. One can express this condition with aid of a new variable y_{kji} that shall be 1 if job j is starting before i on machine M_k and 0 else. Therefore let $y_{kij} \in \mathcal{N}$ and $0 \leq y_{kij} \leq 1$. Next define the number $T_\infty := \sum_{k,i} p_{ki} \geq \sup_{i,j} |x_{li} - x_{lj}| \forall l$. Then we have the conditions:

$$(T_\infty + p_{ki})y_{kji} + (x_{kj} - x_{ki}) \geq p_{ki} \quad \forall i, j, k \tag{8}$$
$$(T_\infty + p_{kj})y_{kji} + (x_{ki} - x_{kj}) \geq p_{kj} \quad \forall i, j, k \tag{9}$$

Note that the coditions imply that if $(x_{ki} - x_{kj}) > 0$ and (8) holds then $y_{kji} = 0$ and on the other side if $(x_{ki} - x_{kj}) \leq 0$ and (9) hold then $y_{kji} = 1$, as required. Further we see that the two conditions ensure that start times can not overlap. To ensure that a job j is executed in its right order $M_{j(1)}, \ldots, M_{j(m)}$ we introduce the variable z_{hkj} that is 1 if the job j needs to be executed on machine M_h before it can be executed on M_k and 0 otherwhise. We can precompute the value of z_{hkj} and therefore add the last condition:

$$z_{hkj}(x_{hj} + p_{hj}) \leq x_{kj} \quad \forall j \tag{10}$$

The full MILP is therefore:

$$
\begin{aligned}
&\min C_{max} && (11)\\
(T_\infty + p_{ki})y_{kji} + (x_{kj} - x_{ki}) &\geq p_{ki} \quad \forall i, j, k && (12)\\
(T_\infty + p_{kj})y_{kji} + (x_{ki} - x_{kj}) &\geq p_{kj} \quad \forall i, j, k && (13)\\
z_{hkj}(x_{hj} + p_{hj}) &\leq x_{kj} \quad \forall j && (14)\\
x_{kj} &\geq 0 \quad \forall k, j && (15)\\
ykij &\geq 0 \quad \forall k, i, j && (16)\\
ykij &\leq 1 \quad \forall k, i, j && (17)\\
& && (18)
\end{aligned}
$$

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**)

Answer To model the outcome of the games we create three variables indexed by the games x^w_{ij}, x^d_{ij}, $x^l_{ij} \in \{0, 1\}$ where the first is one is 1 if team i wins against j at home and 0 otherwhide, the second one is 1 if the game is a draw and the third 1 if i looses abd therefore j wins. Note that we only create variables for $i \neq j$ and therefore can model a game at home for team i by the index ij and a game away by ji. A team can either win, loose or have a draw with another team. Therefore we have the constraint:

$$x^w_{ij} + x^d_{ij} + x^l_{ij} = 1 \forall i, j \text{ with } i \neq j \tag{19}$$

The points of a team can be described by the points earned in home games and away games:

$$p_t = \sum_{j, j \neq t} (3x^w_{tj} + x^d_{tj} + 3x^l_{jt} + x^d_{jt}) \tag{20}$$

Note that for an away game of theam t we have the indices jt such that x^w_{jt} is 1 if j wins but x^l_{jt} if one if t wins, which is the reason we include it above. Next we want to order the points from $t = 0$ with the lowest points to $t = n$ that acheaves the highest points. Therefore let $p_t \leq p_{t+1}$ for $t = 0, 1, ..., n-1$. The objective is then to maximize p_k such that we then know that if a team has $_k + 1$ points it is surely not relegated in any season outcome. The MILP is therefore:

$$\max p_k = \max \left[\sum_{j, j \neq k} (3x^w_{kj} + x^d_{kj} + 3x^l_{jk} + x^d_{jk}) \right] \tag{21}$$

$$x^w_{ij} + x^d_{ij} + x^l_{ij} = 1 \quad \forall i, j \text{ with } i \neq j \tag{22}$$

$$\sum_{j, j \neq t} (3x^w_{tj} + x^d_{tj} + 3x^l_{jt} + x^d_{jt}) \leq \sum_{j, j \neq t+1} (3x^w_{t+1j} + x^d_{t+1j} + 3x^l_{jt+1} + x^d_{jt+1}) \quad \forall t \in \{0, 1, ..., n-1\} \tag{23}$$

$$\tag{24}$$

We want to solve the the MILP for $n = 18$ and $k = 3$ by the Python MIP package (link).

Listing 1:

```
import pip
import numpy as np
pip.main([’install’, ’mip’])
from mip import Model, xsum, maximize, BINARY

nt = 18
k  = 2

m = Model("Sports League")

xw = []
xd = []
xl = []

for i in  range(nt):
  xw.append([])
  xd.append([])
  xl.append([])
  for j in  range(nt):
    if i!=j:
      xw[i].append(m.add_var(var_type=BINARY))
      xd[i].append(m.add_var(var_type=BINARY))
      xl[i].append(m.add_var(var_type=BINARY))
      m += (xw[i][j] + xd[i][j] + xl[i][j]) == 1
    else:
      xw[i].append([])
      xd[i].append([])
      xl[i].append([])

for t in  range(nt-1):
  m +=    xsum( 3*xl[i][t]   + xd[i][t]   +
                3*xw[t][i]   + xd[t][i]  for i in  range(nt) if i!=t) <=\
          xsum( 3*xl[i][t+1] + xd[i][t+1] +
                3*xw[t+1][i] + xd[t+1][i]  for i in  range(nt) if i!=(t+1))

m.objective = maximize(xsum( 3*xl[i][k] + xd[i][k] +
                             3*xw[k][i] + xd[k][i]  for i in  range(nt) if i!=k))
m.optimize()

pk = np. sum([3* int(xl[i][k].x) +  int(xd[i][k].x) +
  3* int(xw[k][i].x) +  int(xd[k][i].x) for i in  range(nt) if i!=k])

print("Team k={} has {} points and will still relegate.". format(k,pk)+\
  "Therefore a team is safe if it has at least {} points". format(pk+1))
```

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)

$$\min \sum_{e \in E} w_e x_e \tag{25}$$

$$\text{s.t.} \sum_{e \in E} x_e = n - 1 \tag{26}$$

$$\sum_{e \in C} x_e \leq |C| - 1 \qquad \forall C \subseteq E,\ |C| \geq 2,\ C \text{ forms a cycle} \tag{27}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E \tag{28}$$

- subtour elimination formulation (SUB)

$$\min \sum_{e \in E} w_e x_e \tag{29}$$

$$\text{s.t.} \sum_{e \in E} x_e = n - 1 \tag{30}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \qquad \forall S \subseteq V, S \neq \emptyset \tag{31}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E \tag{32}$$

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 conv(X).

Exercise 7 *Cover Inequality (**1 point**)*

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

(i) Find a minimum cover C and a corresponding valid cover inequality.

(ii) Find the extended cover inequality for C.

(iii) Lift the inequality to obtain a strong valid inequality, i.e., compute all lifting coefficients.

Exercise 8 *Totally Unimodularity (2 **points**)*

Prove of 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 ore 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):

$$\max \sum_{i\in M}\sum_{j\in N} p_{ij}x_{ij} - \sum_{j\in N} f_j y_j \tag{33}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad \forall i \in M \tag{34}$$

$$x_{ij} \le y_j \qquad \forall i \in M, j \in N \tag{35}$$

$$x_{ij} \ge 0 \qquad \forall i \in M, j \in N \tag{36}$$

$$y_j \in \{0, 1\} \qquad \forall j \in N \tag{37}$$

Your task is now to dualize the linking constraints $x_{ij} \le y_j$, $\forall i \in M$, $\forall j \in N$, 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

$$\max\ 10y_1 + 4y_2 + 14y_3 \tag{38}$$

$$3y_1 + y_2 + 4y_3 \le 4 \tag{39}$$

$$y_1, y_2, y_3 \in \{0, 1\}. \tag{40}$$

Dualize the knapsack constraint (39) 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 $L = \{l_1, \dots, l_n\}$, $0 \le l_j \le W$ for $1 \le j \le n$. 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 (41)–(45). In (41)–(45), 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.

$$\min \quad z = \sum_{i=1}^{n} y_i \tag{41}$$

$$\text{s.t.} \sum_{j=1}^{n} l_j x_{ij} \le W y_i \qquad i = 1, \dots, n \tag{42}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \dots, n \tag{43}$$

$$y_i \in \{0,1\} \qquad i = 1 \dots, n \tag{44}$$

$$x_{ij} \in \{0,1\} \qquad i, j = 1 \dots, n \tag{45}$$

- 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 (41)–(45). (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 $c_e \ge 0$, $\forall e \in E$, 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.)
