Problems 1 and 2 are for the first homework. Write linear programs of both homeworks in the canonical and the equation forms. Solutions must be submitted before the next lecture (not tutorial!) to be evaluated. Students are not allowed to keep submitted solutions after evaluation.

Problem 1 (3-partition)**.** The problem is to decide whether a given multiset of integers can be partitioned into triples that all have the same sum. Formulate this problem using Integer Linear Programming.

Problem 2. We are asked to schedule n jobs on m machines so that

- every job must be scheduled on one machine and the processing of a job cannot be interrupted (preempted);
- every job j is given a release time $r_j \in \mathbb{N}$ and its processing cannot start before that time;
- every job j is given a processing time $p_j \in \mathbb{N}$, i.e. how much time is needed to finish the job;
- every job must be finished before time $T \in \mathbb{N}$;
- every job j is given a deadline $d_j \in \mathbb{N}$.

We have to pay a fee for every job which is finished after its deadline and the fee is the sum of delay of late jobs. Formulate this problem using Integer linear programming.

Problem 3. Plan a production of chocolate for the next year so that the total cost is minimal. The predicted demand of chocolate during the i-th month is d_i units. The change of the production between two consecutive month cost 1500 CZK per unit. Storing chocolate from one month to the following one cost 600 CZK per unit. Chocolate can be stored at most one month because shelf life. As usually, formulate this problem using linear programming.

Is it necessary to consider the production cost?

Problem 4 (Sudoku)**.** Sudoku can be easily solved using a backtrack. Is it also possible to solve it using Linear programming?

Problem 5. Write the following problems both in the canonical and the equation forms.

$$\begin{array}{lrcrcl} \text{Maximize} & 2\boldsymbol{x}_1 & - & 3\boldsymbol{x}_2 & & \\ \text{subject to} & 4\boldsymbol{x}_1 & - & 5\boldsymbol{x}_2 & \leq & 6 \\ & 7\boldsymbol{x}_1 & + & 8\boldsymbol{x}_2 & = & 8 \\ & \multicolumn{5}{l}{\boldsymbol{x}_1 \geq 0.} \end{array}$$

$$\begin{array}{ll} \text{Maximize} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{subject to} & A'\boldsymbol{x} \geq \boldsymbol{b}' \\ & A''\boldsymbol{x} = \boldsymbol{b}'' \\ & \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{x} \geq 0 \end{array}$$

where$A' \in \mathbb{R}^{m' \times n},\ A'' \in \mathbb{R}^{m'' \times n}$, $\boldsymbol{b}' \in \mathbb{R}^{m'},\ \boldsymbol{b}'' \in \mathbb{R}^{m''},\ \boldsymbol{c} \in \mathbb{R}^n$.

Problem 6. Suppose we have a system of linear inequalities that also contains strict inequalities. One that may look like this:

$$\begin{array}{rcrcrcrcr} 5x & + & 3y & & & & & \leq & 8 \\ 2x & & & - & 5z & & & < & -3 \\ 6x & + & 5y & & & + & 2w & = & 5 \\ & & & & 3z & + & 2w & > & 5 \\ & & & & \multicolumn{3}{r}{x, y, z, w} & \geq & 0 \end{array}$$

Is there a way to check if this system has a feasible solution using a linear program?

Does this mean that linear programming allows strict inequalities? Not really. As a strange example, construct a “linear program with a strict ineqality” that satisfies the following:

- There is a simple finite upper bound on its optimum value;
- there is a feasible solution; and
- there is no optimal solution.

This may not happen for a linear program – for a bounded LP, once there exists a feasible solution, there exists also an optimal solution.

Problem 7. Using the graphical methods find the optimal solutions of two objective functions

- min $\boldsymbol{x}_1 + \boldsymbol{x}_2$
- max $\boldsymbol{x}_1 + \boldsymbol{x}_2$

subject to the following conditions.

$$\begin{pmatrix} 1 & 3 \\ 1 & 0 \\ 3 & -1 \\ -2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} \geq \begin{pmatrix} 14 \\ 0 \\ 0 \\ -7 \\ 8 \end{pmatrix}$$

Problem 8. A steel factory produces wire ropes of length L which is bought by an electricity distributor to reinforcement various parts of the grid. The distributor estimates that cables of length $a_1, \dots, a_n$ are needed for the reinforcement (assume that $a_1, \dots, a_n \leq L$). So, the question is how to cut wire ropes of length L into cables of length $a_1, \dots, a_n$ so that the sum of length of unused parts is minimal. Model this problem using a single Integer linear programming problem.

Problem 9. Write a linear programming program which decides whether a given oriented graph contains an oriented cycle.

Problem 10. The travelling salesperson problem (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

Formulate the Travelling salesman problem using (integer) linear programming.

Problem 11. The k-colouring problem of a graph (V, E) is assigning one of k colours to every vertex such that no two adjacent vertices share the same colour. Using Integer linear programming determine whether a given graph is k-colourable for a given integer k.

Problem 12. A square at the center in the point (r_1, r_2) with sides of length s is the square with vertices $\{(r_1 + s_1, r_2 + s_2);\ s_1, s_2, \in \{-s/2, s/2\}\}$.

An input of the problem is a set of points $b_1, \dots, b_n$. For every point b_i, we have to assign a square at the center b_i with sides of length $a_i >= 0$. The area of the intersection of every pair of squares has to be zero, i.e. the intersection has to be at most an side. For instance, you can assign to points $(0, 0)$ and $(5, 4)$ squares with sides of length 2 and 1, or 6 and 4, or 10 and 0; however, sides of length 4 and 7 are forbidden. If a square has a side of length 0, then it is allowed to lie on a side (vertex) of a square but it must not lie inside another square.

Model this problem using linear programming.

Problem 13 (The Classic Transportation Problem I)**.** An unknown country has n bakeries and m shops. Every day, the i-th bakery bakes b_i breads and the j-the shop sells s_j breads. The transportation of one bread from the i-th bakery to the j-th shop cost $c_{i,j}$. Describe the following problem using linear programming: Find the cheapest transportation of all breads so that all breads are transported according to the number of baked and sold breads.

Furthermore, find necessary and sufficient conditions for the existence of a feasible and an optimal solutions.

Problem 14 (The Classic Transportation Problem II)**.** It turned out that whenever i-th bakery supplies j-th shop (with positive number of breads), logistic is needed to organize the transportation which cost an extra $l_{i,j}$. Using linear programming formulate the problem of minimizing the total transportation and logistic expenses.