NUMERICAL OPTIMIZATION

Sheet 2: Linear Programming

Exercise one For the first exercise, we return to *positive definiteness*. We have seen that we can check PD using the Cholesky algorithm computing the decomposition of the same name.

If we are checking PD in class or by hand, we can also use the following criterion, known as *Sylvester's criterion*:

T: A symmetric $n \times n$ matrix is positive definite if and only if all of its n *leading principal minors* have a positive determinant.

By *leading principal minor* we mean the n square submatrices that start in the top left corner. So the first one is just the top left corner, the second one is the 2×2 square containing the top left corner, and so on.

Test the following matrix using this criterion:

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

Exercise two

Now consider the slightly different matrix:

$$A = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$$

Do the following:

1. Test the matrix via the Sylvester's criterion above. What do you get?
2. Run the Cholesky algorithm from the previous session. What do you get? And what is the final conclusion about the properties of the matrix?

Exercise three

Formulate the following as a linear program. Use graphical methods (or guesswork) to find the optimal solution.

Ingredients:

	Pizza	Lasagne	available
Tomatoes	2	3	18
Cheese	4	3	24

Profit: Pizza 16 PLN, Lasagne 14 PLN

Task: Determine the optimal number (possibly fractional) of pizza and lasagne to maximize total profit.

Exercise four Formulate the Knapsack Problem as an integer linear program. On input, we have n items, each item i with weight w_i and price p_i. We have a specified knapsack with weight capacity H and we are trying to maximize the profit we carry in our backpack while staying below the capacity H.

Exercise five Formulate an integer linear program for Minimum cost perfect matching in bipartite graphs:

On input, you have a edge-weighted bipartite graph $G = (U \cup V, E, w)$. For your reminder, a bipartite graph consists of two disjoint sets of vertices (parts) U and V, and its defining feature is that all the edges of the graph have exactly one endpoint in U and one endpoint in V.

As stated above, the graph is edge-weighted, so for any edge $e \in E$ we have a non-negative weight w_e. The goal of Minimum Cost Perfect Matching is to find a collection of edges $M \subseteq E$ such that every vertex participates in *exactly* one edge in M, and the total sum of weights of M is minimized.

Exercise six

Model the following problem as a *linear* program (so no binary or integer variables):

A cargo plane has three compartments for storing cargo: front, center and rear. These compartments have the following limits on both weight and space:

Compartment	Weight capacity (tons)	Space capacity (m^3)
Front	10	6800
Center	16	8700
Rear	8	5300

Furthermore, the weight of the cargo in the respective compartments must be the same proportion of that compartment's weight capacity to maintain the balance of the plane.

The following four cargoes are available for shipment on the next flight:

Cargo	Weight (tons)	Volume (m^3/ton)	Profit (EUR/ton)
C1	18	480	310
C2	15	650	380
C3	23	580	350
C4	12	390	285

Any proportion of these cargos can be accepted. The objective is to determine how much (if any) of each cargo C1, C2, C3 and C4 should be accepted and how to distribute each among the compartments so that the total profit for the flight is maximised.