Technische Universität München, Zentrum Mathematik
Lehrstuhl für Angewandte Geometrie und Diskrete Mathematik

Discrete Optimization (MA 3502)

Prof. Dr. P. Gritzmann | Dipl.-Math. Viviana Ghiglione | Dr. M. Ritter

Exercise Sheet 2

Exercise 2.1 (The Euclidean Algorithm) [8 credits]
Consider the following *Euclidean Algorithm*:

Input: $a, b \in \mathbb{N}$
Output: $\gcd(a, b)$
$k \leftarrow 0$;
$\sigma_0 \leftarrow \max\{a, b\}$;
$\sigma_1 \leftarrow \min\{a, b\}$;
repeat
  $k \leftarrow k + 1$;
  $\sigma_{k+1} \leftarrow \sigma_{k-1} - \left\lfloor \frac{\sigma_{k-1}}{\sigma_k} \right\rfloor \sigma_k$;
until $\sigma_{k+1} = 0$;
return σ_k

a) Let $a, b \in \mathbb{N}$ with $a \geq b \geq 1$. Show that $\lfloor \frac{a}{b} \rfloor \cdot b \geq \frac{a}{2}$.

b) Prove that the Euclidean Algorithm returns a correct solution in at most $\lfloor \log_2(a \cdot b) \rfloor + 1$ iterations.

c) Show that the Euclidean Algorithm can be modified to return $x_1, x_2 \in \mathbb{Z}$ with $a \cdot x_1 + b \cdot x_2 = \gcd(a, b)$.

Hint: Construct two sequences α_k and β_k, $k \in \mathbb{N}$, with $a \cdot \alpha_k + b \cdot \beta_k = \sigma_k$.

d) Find two integers x_1, x_2 with $121x_1 + 19x_2 = 1$.

Answer to Exercise 2.1

a) For $a \geq b \geq 1$ we have $2 \lfloor \frac{a}{b} \rfloor = \lfloor \frac{a}{b} \rfloor + \lfloor \frac{a}{b} \rfloor \geq (\frac{a}{b} - 1) + 1 = \frac{a}{b}$.

b) Note that $\sigma_{k-1} - \left\lfloor \frac{\sigma_{k-1}}{\sigma_k} \right\rfloor \sigma_k$ is the *remainder* in dividing σ_{k-1} by σ_k.

Correctness We start by proving correctness of the algorithm: Let $a \geq b$. Any divisor $\kappa \in \mathbb{N}$ of a and b clearly divides $a - \lfloor \frac{a}{b} \rfloor b = \sigma_2$. On the other hand, if κ divides $\sigma_2 = a - \lfloor \frac{a}{b} \rfloor b$ and b, then κ divides also a. In other words,

$$\gcd(a, b) = \gcd\left(a - \left\lfloor \frac{a}{b} \right\rfloor b, b\right) = \gcd(\sigma_2, \sigma_1).$$

It is easily verified that $\sigma_2 < \sigma_1$, and we thus obtain a decreasing sequence $\sigma_2, \ldots, \sigma_n$ with $\gcd(a, b) = \gcd(\sigma_2, \sigma_1) = \cdots = \gcd(\sigma_{n-1}, \sigma_n)$. The algorithm terminates if and only if $\sigma_{n+1} = 0$, and in that case clearly the correct $\gcd(\sigma_{n-1}, \sigma_n) = \gcd(a, b)$ is returned.

Iterations (and Finiteness) According to a) we have

$$\sigma_{k+1} = \sigma_{k-1} - \left\lfloor \frac{\sigma_{k-1}}{\sigma_k} \right\rfloor \sigma_k \leq \sigma_{k-1} - \frac{\sigma_{k-1}}{2} = \frac{\sigma_{k-1}}{2}.$$

As $\sigma_k \in \mathbb{N}$ for every k, the algorithm will eventually yield $\sigma_k = 0$ and therefore terminate. More precisely, after the first iteration we have reduced $\sigma_0 = a$ by a factor of 2, after the second iteration we have reduced $\sigma_1 = b$ by a factor of 2, and so on. Thus, we need at most $\lfloor \log_2 a + \log_2 b \rfloor + 1 = \lfloor \log_2(ab) \rfloor + 1$ iterations to reduced one of the two numbers to 0 and have the algorithm terminate. Each iteration includes two assignments, a division, a multiplication, an addition, and a subtraction. In summary, the running time of the algorithm is linear in the number of iterations and thus linear in terms of the encoding-length.

c) Consider the following extension of the Euclidean Algorithm:

Input: $a, b \in \mathbb{N}$
Output: $\gcd(a,b)$, $x_1, x_2 \in \mathbb{Z}$ with $\gcd(a,b) = x_1 a + x_2 b$
$k \leftarrow 0$;
$\sigma_0 \leftarrow \max\{a,b\}$;
$\sigma_1 \leftarrow \min\{a,b\}$;
$\alpha_0 \leftarrow 1$, $\beta_0 \leftarrow 0$;
$\alpha_1 \leftarrow 0$, $\beta_1 \leftarrow 1$;
repeat
  $k \leftarrow k+1$;
  $\sigma_{k+1} \leftarrow \sigma_{k-1} - \left\lfloor \frac{\sigma_{k-1}}{\sigma_k} \right\rfloor \sigma_k$;
  $\alpha_{k+1} \leftarrow \alpha_{k-1} - \left\lfloor \frac{\sigma_{k-1}}{\sigma_k} \right\rfloor \alpha_k$;
  $\beta_{k+1} \leftarrow \beta_{k-1} - \left\lfloor \frac{\sigma_{k-1}}{\sigma_k} \right\rfloor \beta_k$;
until $\sigma_{k+1} = 0$;
return $\sigma_k, \alpha_k, \beta_k$

The initialization gives us

$$\begin{aligned} \sigma_0 = a = 1 \cdot a + 0 \cdot b = \alpha_0 \cdot a + \beta_0 \cdot b \\ \text{and} \quad \sigma_1 = b = 0 \cdot a + 1 \cdot b = \alpha_1 \cdot a + \beta_1 \cdot b, \end{aligned}$$

and inductively, with $q := \left\lfloor \frac{\sigma_{k-1}}{\sigma_k} \right\rfloor$, we have

$$\begin{aligned} \sigma_{k+1} &= \sigma_{k-1} - q\sigma_k \\ &= \alpha_{k-1}a + \beta_{k-1}b - q(\alpha_k a + \beta_k b) \\ &= a(\alpha_{k-1} - q\alpha_k) + b(\beta_{k-1} - q\beta_k) \\ &= a\alpha_{k+1} + b\beta_{k+1}, \end{aligned}$$

showing the correctness of the algorithm.

d) With the extended Euclidean Algorithm we obtain $121 \cdot (-8) + 19 \cdot 51 = 1$ with the following calculations:

$\mathbf{k}$	$\sigma_\mathbf{k}$	$\alpha_\mathbf{k}$	$\beta_\mathbf{k}$
0	121	1	0
1	19	0	1
2	7	1	-6
3	5	-2	13
4	2	3	-19
5	1	-8	51

Exercise 2.2 [4 credits]

In Sudoku a 9×9 square is given, which is divided into nine 3×3 boxes. Some entries contain prescribed numbers from 1 to 9. The task is to fill the remaining entries such that within each row, column, and box every number from 1 to 9 appears exactly once.

Model the Sudoku task as an integer linear optimization problem.

Answer to Exercise 2.2

Our model contains $9^3 = 729$ binary variables $\xi_{ijk} \in \{0, 1\}$, $i, j, k \in [9]$. A value of $\xi_{ijk} = 1$ means that k is the entry at position (i, j) of the Sudoku grid; the value $\xi_{ijk} = 0$ indicates that k is not an entry at (i, j). Let $F \subset [9]^2$ denote the positions in the Sudoku grid that are prescribed by a number $k(i, j)$. Moreover, let $B_l \subset [9]^2$, $l \in [9]$ denote the positions in the l-th box. Then the following linear constraints model the Sudoku task:

$$\sum_{i=1}^{9} \xi_{ijk} = 1, \quad j, k = 1, \ldots, 9 \tag{1}$$

$$\sum_{j=1}^{9} \xi_{ijk} = 1, \quad i, k = 1, \ldots, 9 \tag{2}$$

$$\sum_{(i,j) \in B_l} \xi_{ijk} = 1, \quad k, l = 1, \ldots, 9 \tag{3}$$

$$\sum_{k=1}^{9} \xi_{ijk} = 1, \quad i, j = 1, \ldots, 9 \tag{4}$$

$$\xi_{ij,k(i,j)} = 1, \quad (i, j) \in F. \tag{5}$$

Conditions (1), (2), and (3) ensure that every number occurs exactly once in each row, column, and box. Condition (4) ensures that each field contains an entry while condition (5) deals with the prescribed entries.

Exercise 2.3 [6 credits]

Consider the Traveling Salesman Problem (TSP) on the complete directed graph $D_n = (V, A)$ on the vertex set $V = \{0, 1, \ldots, n-1\}$ with some distance function $d : A \to \mathbb{R}_{\geq 0}$. In the following, we will devise integer programming models for the TSP using variables $x_a \in \{0, 1\}$ for every arc $a \in A$ that indicate whether a is used in the optimal tour ($x_a = 1$) or not ($x_a = 0$).

a) Consider the following integer linear program:

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in A} d_{ij}x_{ij} \\
& \sum_{j\in V:(i,j)\in A} x_{ij} = 1 \quad \text{for all } i \in V \\
& \sum_{i\in V:(i,j)\in A} x_{ij} = 1 \quad \text{for all } j \in V \\
& x_{ij} \in \{0,1\} \quad \text{for all } (i,j) \in A
\end{aligned}$$

Show that this ILP is not sufficient to capture the TSP by providing a feasible solution to the ILP that does not correspond to a feasible TSP tour.

b) Can you add additional linear constraints to the above ILP such that the feasible set corresponds to the set of feasible TSP tours? How many variables and how many constraints do you need for your formulation?

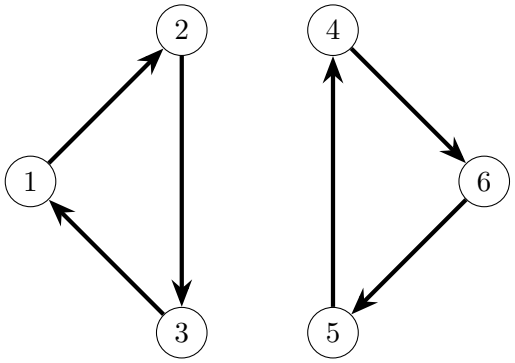
c) Consider the following mixed integer linear program:

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in A} d_{ij}x_{ij} \\
& \sum_{j\in V:(i,j)\in A} x_{ij} = 1 \quad \text{for all } i \in V \\
& \sum_{i\in V:(i,j)\in A} x_{ij} = 1 \quad \text{for all } j \in V \\
& u_i - u_j + (n-1)x_{ij} \leq n-2 \quad \text{for all } (i,j) \in A \text{ with } i,j \neq 0 \\
& x_{ij} \in \{0,1\} \quad \text{for all } (i,j) \in A \\
& u_i \in \mathbb{R} \quad \text{for all } i \in V \setminus \{0\}
\end{aligned}$$

Show that the feasible set of this MILP corresponds to the set of feasible traveling salesman tours.

Answer to Exercise 2.3

a) Consider the complete directed graph on 6 vertices and the "solution" depicted below. While this is certainly not a feasible TSP tour, it does correspond to a feasible solution of the ILP given in the problem description.



b) To prevent situations with *subtours* (such as the one shown above), we need to add constraints that render these subtours infeasible. One possibility to do that is to eliminate cycles with less than n vertices by adding the following set of constraints known as *subtour elimination constraints*:

$$\sum_{i,j \in S:(i,j) \in A} x_{ij} \leq |S| - 1 \quad \text{for all } S \in \mathcal{P}(V) \setminus \{\emptyset, V\}$$

This yields a valid ILP formulation of the Traveling Salesman Problem. Unfortunately, the number of constraints needed in that formulation is $2^n - 2$, an exponential number, which makes it impractical to handle. (Note: In applications involving an ILP model for TSP this formulation is commonly used. The trick is to start the solution process without the subtour elimination constraints, then generate constraints for some of the subtours that pop up and re-solve. This is repeated until a subtour-free solution is obtained. Of course, the important thing here is to find violated subtour inequalities efficiently—this topic will be covered in the lecture *Combinatorial Optimization*.)

c) We first show that every feasible solution of our MILP corresponds to a feasible tour. We start by proving that every cycle that is part of the solution passes through node 0: Suppose there was a cycle $v_1, v_2, \ldots, v_k$ of nodes that does not include node 0. Then we can write down the inequalities along the arcs $(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k), (v_k, v_1)$ along the tour:

$$\begin{aligned}
&u_1 - u_2 + (n-1)x_{12} \leq n - 2 \\
&u_2 - u_3 + (n-1)x_{23} \leq n - 2 \\
&\vdots \\
&u_{k-1} - u_k + (n-1)x_{k-1,k} \leq n - 2 \\
&u_k - u_1 + (n-1)x_{k1} \leq n - 2
\end{aligned}$$

Adding all these constraints and substituting the arc variables with the value 1 yields

$$k(n-1) \leq k(n-2),$$

clearly a contradiction for any $k > 0$. Thus, every cycle in the solution contains the node 0. As there can only be one in-arc and one out-arc for that node, the solution consists of just one cycle, so it is in fact a feasible TSP tour.

For the reverse direction, let $\tau = (v_0, v_1, \ldots, v_{n-1})$ be a tour with $v_0 = 0$ passing through the nodes $0 = v_0, v_1, \ldots$ in that order. We will of course set $x_{v_0,v_1} = x_{v_1,v_2} = \ldots = x_{v_{n-1},1} = 1$ and all the other x-variables to 0, but we still have to provide feasible values for the u-variables. To do so, we set $u_{v_1} = 1$, $u_{v_2} = 2$ and so on, hence assigning to each node a u-value corresponding to its place in the tour if we start at node 0.

First consider an arc $(v_i, v_j) \in A$ that is not part of the tour, thus $x_{ij} = 0$. The corresponding constraint is then

$$u_i - u_j \leq n - 2,$$

which is always valid, because the case $u_i = n - 1$ and $u_j = 0$ would only occur for the arc $(v_{n-1}, 0)$, which is part of the tour (and thus $x_{v_{n-1},0} = 1$).

If, on the other hand, we consider some arc $(v_i, v_j) \in A$ with $x_{ij} = 1$ we know that $u_i - u_j = -1$, because the nodes v_i and v_j are successive nodes on the tour. (Note that there is no inequality for the arc (v_{n-1}, v_0)!) Hence the inequality becomes

$$-1 + (n-1) \leq n - 2,$$

which is trivially fulfilled. This show that the proposed setting of the variables is indeed valid, so a feasible solution corresponds to every tour and vice versa. (Note: This setting is by no means unique, so a different idea might also get you a valid proof. Also, the interpretation of the u_i-variables as "position in the tour, starting at node 0" will not be valid for a different solution that could correspond to the same tour.)

Exercise 2.4 [5 credits]

a) Let $a \in \mathbb{Z}^n$ and $\beta \in \mathbb{Z}$. Show that the linear diophantine equation

$$a^T x = \beta$$

has an integer solution $x \in \mathbb{Z}^n$ if and only if $\gcd(a_1, a_2, \ldots, a_n)$ divides β.

b) Show that the linear diophantine equation $121x_1 + 19x_2 = \beta$, has an integer solution $x \in \mathbb{Z}^2$ for every $\beta \in \mathbb{Z}$. Determine all solutions by the method discussed in the lectures, and interpret your result geometrically.

Answer to Exercise 2.4

a) We start with a simple argument that uses knowledge from exercise 2.1: If the diophantine equation has an integer solution $x^* \in \mathbb{Z}^n$, then we have $a^T x^* = \beta$. Of course, the $\gcd(a_1, \ldots, a_n)$ divides each summand on the left hand side, and thus it also divides β.

On the other hand, let $\kappa := \gcd(a_1, \ldots, a_n)$ and assume κ divides β. From 2.1 we know there are integers $x'_1, \ldots, x'_n$ such that $a^T x' = \kappa$. Let $\gamma := \frac{\beta}{\kappa} \in \mathbb{Z}$, then $a^T(\gamma x') = \gamma\kappa = \beta$, thus $x^* = \gamma x'$ is an integer solution of the linear diophantine equation.

For the sake of completeness, we also state a different proof that does not rely on Euclid's algorithm explicitly (but requires a bit more work).

For $a = 0$ the statement is obviously true, hence we assume $a \neq 0$ in the following. Without loss of generality, let $a_1 \neq 0$ (otherwise permute the coordinates suitably). Let

$$S := \left\{\sigma \in \mathbb{N} : \exists x \in \mathbb{Z}^n \text{ with } a^T x = \sigma\right\}$$
$$\text{and} \quad \sigma^* := \min S$$

(By $\mathbb{N}$ we actually mean the set of positive integers, *not* including 0. The set $\mathbb{N} \cup \{0\}$ would be denoted by $\mathbb{N}_0$.) Note that this minimum does indeed exist, as $|a_1| \in S$ and thus $S \neq \emptyset$. Let $x^* \in \mathbb{Z}^n$ such that $a^T x^* = \sigma^*$. As every integral linear combination of the $a_1, a_2, \ldots, a_n$ (and hence σ^*) can be divided by $\gcd(a_1, a_2, \ldots, a_n)$ we know that $\gcd(a_1, a_2, \ldots, a_n) \leq \sigma^*$.

To show that $\gcd(a_1, a_2, \ldots, a_n) \geq \sigma^*$ also holds, we will prove that σ^* divides each of the a_i. That would mean that σ^* also divides $\gcd(a_1, a_2, \ldots, a_n)$ and thus the desired inequality. So consider some $i \in [n]$, then a_i can be represented as

$$a_i = m_i \cdot \sigma^* + r_i$$

with $m_i, r_i \in \mathbb{Z}$ and $0 \leq r_i \leq \sigma^* - 1$ (division with remainder r_i). Thus,

$$r_i = a_i - m_i \cdot \sigma^* = a_i - m_i \cdot (a_1 x_1^* + \cdots + a_n x_n^*),$$

which means that r_i is an integral linear combination of $a_1, \ldots, a_n$ and hence $r_i \in S \cup \{0\}$. With $r_i < \sigma^*$ and σ^* being the minimum of S, the only remaining possibility is $r_i = 0$, hence σ^* divides a_i. As the choice of i was arbitrary, this proves that σ^* is a divisor of $a_1, \ldots, a_n$ and hence the desired inequality.

b) From the Euclidean Algorithm in problem 2.1 we already know

$$(121, 19) \to (7, 19) \to (7, 5) \to (2, 5) \to (2, 1) \to (0, 1),$$

therefore $\gcd(121, 19) = 1$. The linear diophantine equation $121x_1 + 19s_2 = \beta$ is thus solvable for every $\beta \in \mathbb{Z}$. The transitions in the Euclidean Algorithm can be formulated by means of matrix multiplications

$$(121, 19) \xrightarrow{C_1} (7, 19) \xrightarrow{C_2} (7, 5) \xrightarrow{C_3} (2, 5) \xrightarrow{C_4} (2, 1) \xrightarrow{C_5} (0, 1)$$

with

$$C_1 := \begin{pmatrix} 1 & 0 \\ -6 & 1 \end{pmatrix}, C_2 := \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}, C_3 := \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}, C_4 := \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}, C_5 := \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}.$$

Thus, with

$$C := C_1C_2C_3C_4C_5 = \begin{pmatrix} 19 & -8 \\ -121 & 51 \end{pmatrix}$$

we obtain $(121, 19)C = (0, 1)$. (Note that $\det(C_i) = 1$, $i = 1, \ldots, 5$.)

With $y := C^{-1}x$ and every solution x we thus have

$$\beta = (121, 19)x = (121, 19)Cy = (0, 1)y = y_2.$$

The set of solutions of the linear diophantine equation $121x_1 + 19x_2 = \beta$ is therefore given by

$$x = Cy = C \begin{pmatrix} y_1 \\ \beta \end{pmatrix} = \begin{pmatrix} 19 \\ -121 \end{pmatrix} y_1 + \begin{pmatrix} -8 \\ 51 \end{pmatrix} \beta$$

with $y_1 \in \mathbb{Z}$.

The set of solutions can be viewed as a lattice $L := \left\{(19, -121)^T y_1 : y_1 \in \mathbb{Z}\right\}$, which is translated by the vector $\beta(-8, 51)^T$.