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

Combinatorial Optimization (MA 4502)

Prof. Dr. A. Taraz | Dr. M. Ritter

exercise sheet 1

Exercise 1.1 (Duality—a little reminder)
Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c, x \in \mathbb{R}^n$ and consider the following linear programming problem and its corresponding dual:

$$\begin{aligned} \max\ & c^T x \\ & Ax \leq b \end{aligned} \qquad \text{(P)} \qquad\qquad \begin{aligned} \min\ & b^T y \\ & A^T y = c \\ & y \geq 0 \end{aligned} \qquad \text{(D)}$$

A basis $B \subset \{1, \dots, m\}$ is called *primal feasible*, if $x := A_B^{-1} b_B$ is a feasible solution of (P). It is called *dual feasible*, if $y \in \mathbb{R}^m$ defined through

$$y_B := (A_B^T)^{-1} c, \qquad y_N := 0 \quad (\text{where } N := \{1, \dots, m\} \setminus B)$$

is a feasible solution of the dual linear program.

a) Assume you solve the dual problem using the simplex algorithm. What information do you get about the problem (primal and/or dual) and its solution?

b) Give a geometric interpretation of the constraints and the objective function of the dual problem in the primal problem's geometric space.

c) Give a geometric interpretation of primal feasible and dual feasible basis.

d) Sketch a two-dimensional polyhedron where you can find

   i) a primal feasible basis that is not dual feasible

   ii) a dual feasible basis that is not primal feasible

   Can you also find an example of a basis that is both primal and dual feasible? Is there on with a basis that is neither primal feasible nor dual feasible?

e) Sketch an idea for an algorithm that starts with a dual feasible basis and constructs an optimal vertex of (P). What disadvantages might such an algorithm have?

f) Consider the simplex algorithm again and assume you want to add additional constraints after obtaining a solution (like you would in cutting plane algorithms). How does the simplex algorithm handle that situation? What about your algorithm?

g) Assume $m \gg n$. Use the results from the preceding parts of this exercise to construct an algorithm that determines an optimal vertex of (P) (and that can be faster than the standard simplex algorithm).

Please turn over.

Exercise 1.2

In the lecture you have proved that the *minimum spanning tree* heuristic (MST heuristic) is a 2-approximation for metric TSP. Give an example that shows that this bound is asymptotically tight, i. e. construct a family of TSP instances, each with an MST solution, such that the approximation ratio of these solutions converges to 2 for large vertex numbers.

Exercise 1.3 (Nearest Neighbour heuristic)

In this exercise we will look at a common heuristic for the construction of a traveling salesman tour, the *Nearest-Neighbour* (or *NN*) heuristic. Throughout the exercise, assume we have an instance of TSP given by a complete graph $K_n = (V, E)$ on $n \in \mathbb{N}$ vertices with edge weights $c \geq 0$. We will use the notation $\tau = (v_1, v_2, \ldots, v_k)$ for a (sub-)tour connecting the nodes v_1, v_2, …, v_k and v_1 in that order and identify τ with the set of vertices contained in τ; we call v_1 the start and v_k the end node of such a (sub-)tour.

The *Nearest Neighbour* heuristic constructs a feasible tour as follows:

1) Randomly choose a start node $v_1 \in V$, set the current node $v^* := v_1$ and the current subtour $\tau := (v_1)$.
2) Repeat the following steps until all nodes are contained in τ:
   i) Among all nodes in $V \setminus \tau$ find some node v' with minimal distance to the current node v^*:
   $$c(v', v^*) = \min\{c(v, v^*) : v \in V \setminus \tau\}$$
   ii) Add the node v' to the tour $\tau := (v_1, \ldots, v_k = v^*, v')$ and set the current node $v^* := v'$.

a) Show that there is no constant factor $r > 0$ such that the Nearest Neighbour heuristic is an r-approximation for TSP.

In the following, we will only consider metric TSP, i. e., the edge weight function c fulfills the triangle inequality. We recursively define a graph G_k as follows (see fig. 1):

- $G_1 = K_3$ is a triangle on the three vertices l_1, r_1, m_1 where each edge has length 1.
- For $k \geq 2$ the graph G_k consists of two copies $G^L_{k-1} = (V^L_{k-1}, E^L_{k-1})$ and $G^R_{k-1} = (V^R_{k-1}, E^R_{k-1})$ of G_{k-1} and one extra vertex m_k; the nodes l^L_{k-1} and r^R_{k-1} are renamed to l_k and r_k. The edge set E_k consists of $E^L_{k-1} \cup E^R_{k-1}$ and the new edges (r^L_{k-1}, l^R_{k-1}), (r^L_{k-1}, m_k) and (l^R_{k-1}, m_k).

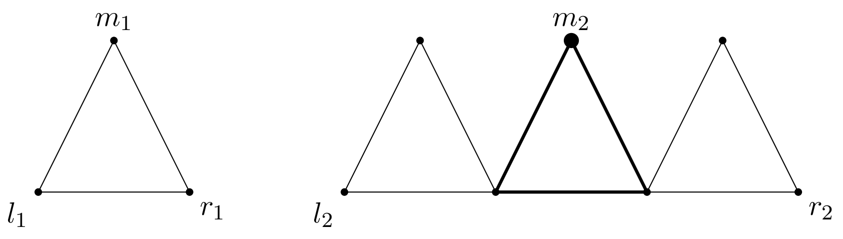

Figure 1: G_1 and G_2.

b) Prove the following lemma. (Hint: Use induction on k.)

. . . continued on next page . . .

Lemma 1
Let $k \geq 1$ and let G' be a graph that contains G_k as an induced subgraph such that all edges between G_k and $G' - G_k$ are incident to either l_k or r_k in G_k. Obtain an instance of metric TSP (G, c) from G' by assigning a length of 1 to every edge in G' and the length of a shortest u-v-path in G' to every edge (u, v) that is not in G'. Then for suitable starting point (and suitable tie-breaking) at some point in the Nearest Neighbour heuristic there exists a partial tour with the following properties:

i) The tour visits exactly the nodes in G_k.

ii) The tour starts in node l_k and ends in node m_k.

iii) The tour has exactly length $(k + 3) \cdot 2^{k-1} - 2$.

c) Use the lemma to prove that for every $k \geq 1$ there is a metric TSP instance with $n = 2^{k+1}$ vertices and an optimal tour of length 2^{k+1} for which the Nearest Neighbour heuristic may yield a tour of length $(k + 4) \cdot 2^{k-1}$.

d) Use your result to prove that Nearest Neighbour does not provide a constant factor approximation algorithm for metric TSP.

Exercise 1.4 (Inapproximability of TSP)
Show that there does not exist an r-approximation algorithm for TSP for any constant factor r, unless $\mathcal{P} = \mathcal{NP}$. (Hint: Use a construction similar to the one in the lecture for proving $\mathcal{NP}$-hardness of TSP.)

This exercise sheet will be discussed in the tutorials on Oct 24 and Oct 25.