École Polytechnique Fédérale de Lausanne
Prof. Eisenbrand

March 11, 2025

Discrete Optimization (Spring 2025)

Assignment 4

1) Let $P = \{x : Ax \leqslant b\} \subseteq \mathbb{R}^n$ be a polyhedron. Show that x^* is an extreme point $\iff \forall x_1 \neq x_2 \in P, x^* \neq \frac{1}{2}x_1 + \frac{1}{2}x_2$.

   Solution:
   We show x^* is an extreme point $\implies \forall x_1 \neq x_2 \in P, x^* \neq \frac{1}{2}x_1 + \frac{1}{2}x_2$.

   Since x^* is an extreme point, there exists an inequality $a^T x \leqslant \beta$ valid for K such that $\{x^*\} = P \cap \{x \in \mathbb{R}^n : a^T x = \beta\}$. Assume that x^* can be written as a midpoint of two points $x_1, x_2 \in K$. We obtain that

   $$\beta = a^T x^* = a^T(1/2x_1 + 1/2x_2) = 1/2a^T x_1 + 1/2a^T x_2 < 1/2\beta + 1/2\beta = \beta$$

   which gives a contradiction. We used that $a^T x_1 < \beta$ since $a^T x_1 \leqslant \beta(a^T x \leqslant \beta$ is valid for P and $x_1 \in P)$ and $a^T x_1 \neq \beta$ (x^* is the only point in P satisfying $a^T x \leqslant \beta$ with equality).

   We show $\forall x_1 \neq x_2 \in P, x^* \neq \frac{1}{2}x_1 + \frac{1}{2}x_2 \implies x^*$ is an extreme point.

   Let A_{x*}be the a set of valid inequalities for P which are tight at x^*. If $rank(A_{x*}) < n$ then there exists a vector d and $\epsilon \in \mathbb{R} > 0$ such that $ad = 0$ for any vector a of A_{x*} and $x^* \pm \epsilon d \in P$. Since $x^* = 1/2(x^* + \epsilon d) + 1/2(x^* - \epsilon d)$ we have a contradiction. As this cannot hold, here exist n linearly independent vectors that form the rows of some matrix A such that each row is a valid inequality $a_i^T x = b_i$ for P and $Ax^* = b$ at x^*. Let $c = 1^T A$. Then x^* is the unique point x such that $cx = 1^T b$ and $Ax \leqslant b$. Thus x^* is an extreme point.

2) Suppose that the linear program $\max\{c^T x \colon x \in \mathbb{R}^n,\, Ax \leqslant b\}$ is non-degenerate and B is an optimal basis. Show that the linear program has a unique optimal solution if and only if $\lambda_B > 0$.

   Solution:
   Let x^* be the optimal solution corresponding to B and $\lambda_B^T A_B = c^T$ with $\lambda_B > 0$. Suppose that there is another optimal solution x'. This gives

   $$0 = c^T(x^* - x') = \lambda_B^T(A_B x^* - A_B x') = \lambda_B^T(b_B - A_B x').$$

   Since $\lambda_B > 0$ and $b_B - A_B x' \geqslant 0$, we must have $b_B - A_B x' = 0$, hence $x' = A_B^{-1} b_B = x^*$.

   Now, assume that x^* is the unique optimal solution. Assume for the sake of contradiction that λ_B has a zero component $\lambda_j = 0, j \in B$. First we choose the direction $d = (-1)A_B^{-1} e_j$ where e_j is the jth unit vector. Next, we shall determine the step-size $\varepsilon > 0$ such that $x^* + \varepsilon d$ is also an optimal solution and different from x^*, which is a contradiction. Note that for any $\varepsilon > 0$ we have

   $$c^T(x^* + \varepsilon d) = c^T x^* + c^T \varepsilon d = c^T x^* + \varepsilon \lambda_B^T A_B d = c^T x^* - \varepsilon \lambda_B^T e_j = c^T x^* - \varepsilon \lambda_j = c^T x^*.$$

   Thus we only need to choose $\varepsilon > 0$ so that $x^* + \varepsilon d$ is feasible. Consider $K = \{j \in \{1, 2, \ldots, m\} : a_j^T d > 0\}$. If K is empty, then we can take any $\varepsilon > 0$. Otherwise let $\varepsilon = \min_{j \in K} \left\{ \frac{b_j - a_j^T x^*}{a_j^T d} \right\}$. Since the linear program is non-degenerate, $b_j - a_j^T x^* > 0, \forall j \in [m] \setminus B$, hence $\varepsilon > 0$.

3) For each of the following assertion, provide a proof or a counterexample.

   i) An index that has just left the basis B in the simplex algorithm cannot enter in the very next iteration.

   ii) An index that has just entered the basis B in the simplex algorithm cannot leave again in the very next iteration.

Solution:

   i) An index that has left the basis can enter in the very next iteration. An example is a triangle in the plane. Maybe the simplex method does not decide to walk to the neighboring optimal vertex in one step but makes a detour (while improving) via the other vertex. In this case, the inequality that has just left re-enters again.

   ii) We use the fact that Simplex always chooses a direction that augments the objective function. Let B be a feasible basis and let Simplex move from B to $\tilde{B} = B \setminus \{i\} \cup \{j\}$, i.e., i leaves the basis and j enters it. Note that B and $\tilde{B}$ have $n-1$ common indices. Assume that j leaves the basis in the next iteration. Let d and $\tilde{d}$ be the directions that Simplex chooses to move from B to $\tilde{B}$ and away from $\tilde{B}$ respectively. Then, $d \cdot a_k = 0 = \tilde{d} \cdot a_k$ for all $k \in B \setminus \{i\}$. Since the set of vectors $a_k : k \in B \setminus \{i\}$ are n$-$1 linearly independent vectors and $d, \tilde{d} \in \mathbb{R}^n$, this means that d and $\tilde{d}$ are parallel. Since j entered the basis, this means that $a_j^T d > 0$. Since j leaves the basis in the next step, $a_j^T \tilde{d} = -1$. Thus, $d = -w\tilde{d}$ for some $w > 0$. In particular, this means that the Simplex is moving in the opposite direction. Now, due to the choice of the direction in Simplex we know that $c^T d > 0$ and $c^T \tilde{d} > 0$. But this is impossible as $d = -w\tilde{d}$.

4) Consider the following linear program:

$$\begin{aligned} \max \quad & 6a + 9b + 2c \\ \text{subject to } & a + 3b + c \leqslant -4 \\ & b + c \leqslant -1 \\ & 3a + 3b - c \leqslant 1 \\ & a \leqslant 0 \\ & b \leqslant 0 \\ & c \leqslant 0. \end{aligned}$$

Solve the linear program with the Simplex method and initial vertex $(-1, -1, 0)^T$. For each iteration indicate all the parameters including the optimal value and the proof of optimality.

Solution:

iteration	basis	vertex	λ	direction	ϵ	index exchange
1	$\{1,2,6\}$	$(-1,-1,0)$	$(6,-9,5)$	$(3,-1,0)$	$1/3$	$2 \to 4$
2	$\{1,4,6\}$	$(0,-4/3,0)$	$(3,3,-1)$	$(0,1/3,-1)$	$5/2$	$6 \to 3$
3	$\{1,3,4\}$	$(0,-1/2,-5/2)$	$(5/2,1/2,2)$			

so that the last vertex is optimal with objective value $-19/2$.

5) Fill in the blanks to complete the code for the Simplex.py file which runs a simplex algorithm for a non-degenerate LP.