15.081J/6.251J Introduction to Mathematical Programming

Lecture 4: Geometry of Linear Optimization III

1 Outline

Slide 1

1. Projections of Polyhedra
2. Fourier-Motzkin Elimination Algorithm
3. Optimality Conditions

2 Projections of polyhedra

Slide 2

- $\pi_k : \Re^n \mapsto \Re^k$ projects $\boldsymbol{x}$ onto its first k coordinates:

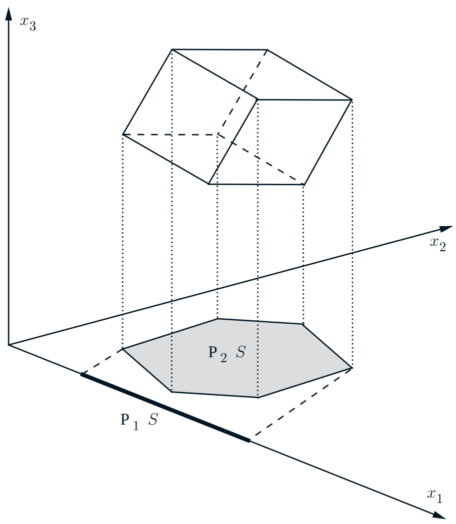
$$\pi_k(\boldsymbol{x}) = \pi_k(x_1, \ldots, x_n) = (x_1, \ldots, x_k).$$

- $$\Pi_k(S) = \left\{\pi_k(\boldsymbol{x}) \mid \boldsymbol{x} \in S\right\};$$

  Equivalently

$$\Pi_k(S) = \Big\{(x_1, \ldots, x_k) \;\Big|\; \text{there exist } x_{k+1}, \ldots, x_n$$

$$\text{s.t. } (x_1, \ldots, x_n) \in S\Big\}.$$



2.1 The Elimination Algorithm

2.1.1 By example

Slide 3

- Consider the polyhedron

$$x_1 + x_2 \;\geq\; 1$$

$$
\begin{aligned}
x_1 + x_2 + 2x_3 &\geq 2 \\
2x_1 + 3x_3 &\geq 3 \\
x_1 - 4x_3 &\geq 4 \\
-2x_1 + x_2 - x_3 &\geq 5.
\end{aligned}
$$

- We rewrite these constraints

$$
\begin{aligned}
0 &\geq 1 - x_1 - x_2 \\
x_3 &\geq 1 - (x_1/2) - (x_2/2) \\
x_3 &\geq 1 - (2x_1/3) \\
-1 + (x_1/4) &\geq x_3 \\
-5 - 2x_1 + x_2 &\geq x_3.
\end{aligned}
$$

- Eliminate variable x_3, obtaing polyhedron Q

$$
\begin{aligned}
0 &\geq 1 - x_1 - x_2 \\
-1 + x_1/4 &\geq 1 - (x_1/2) - (x_2/2) \\
-1 + x_1/4 &\geq 1 - (2x_1/3) \\
-5 - 2x_1 + x_2 &\geq 1 - (x_1/2) - (x_2/2) \\
-5 - 2x_1 + x_2 &\geq 1 - (2x_1/3).
\end{aligned}
$$

2.2 The Elimination Algorithm

SLIDE 4

1. Rewrite $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ in the form

$$
a_{in}x_n \geq -\sum_{j=1}^{n-1} a_{ij}x_j + b_i, \qquad i = 1, \ldots, m;
$$

   if $a_{in} \neq 0$, divide both sides by a_{in}. By letting $\overline{\boldsymbol{x}} = (x_1, \ldots, x_{n-1})$ that P is represented by:

$$
\begin{aligned}
x_n &\geq d_i + \boldsymbol{f}_i'\overline{\boldsymbol{x}}, && \text{if } a_{in} > 0, \\
d_j + \boldsymbol{f}_j'\overline{\boldsymbol{x}} &\geq x_n, && \text{if } a_{jn} < 0, \\
0 &\geq d_k + \boldsymbol{f}_k'\overline{\boldsymbol{x}}, && \text{if } a_{kn} = 0.
\end{aligned}
$$

2. Let Q be the polyhedron in $\Re^{n-1}$ defined by:

$$
\begin{aligned}
d_j + \boldsymbol{f}_j'\overline{\boldsymbol{x}} &\geq d_i + \boldsymbol{f}_i'\overline{\boldsymbol{x}}, && \text{if } a_{in} > 0 \text{ and } a_{jn} < 0, \\
0 &\geq d_k + \boldsymbol{f}_k'\overline{\boldsymbol{x}}, && \text{if } a_{kn} = 0.
\end{aligned}
$$

Theorem:
The polyhedron Q constructed by the elimination algorithm is equal to the projection $\Pi_{n-1}(P)$ of P.

2.3 Implications

SLIDE 5

- Let $P \subset \Re^{n+k}$ be a polyhedron. Then, the set

$$\left\{ \boldsymbol{x} \in \Re^n \;\middle|\; \text{there exists } \boldsymbol{y} \in \Re^k \text{ such that } (\boldsymbol{x}, \boldsymbol{y}) \in P \right\}$$

  is also a polyhedron.

- Let $P \subset \Re^n$ be a polyhedron and let $\boldsymbol{A}$ be an $m \times n$ matrix. Then, the set $Q = \{\boldsymbol{Ax} \mid \boldsymbol{x} \in P\}$ is also a polyhedron.

- The convex hull of a finite number of vectors is a polyhedron.

2.4 Algorithm for LO

SLIDE 6

- Consider $\min \boldsymbol{c'x}$ subject to $\boldsymbol{x} \in P$.
- Define a new variable x_0 and introduce the constraint $x_0 = \boldsymbol{c'x}$.
- Apply the elimination algorithm n times to eliminate the variables $x_1, \ldots, x_n$
- We are left with the set

$$Q = \left\{ x_0 \mid \text{there exists } \boldsymbol{x} \in P \text{ such that } x_0 = \boldsymbol{c'x} \right\},$$

  and the optimal cost is equal to the smallest element of Q.

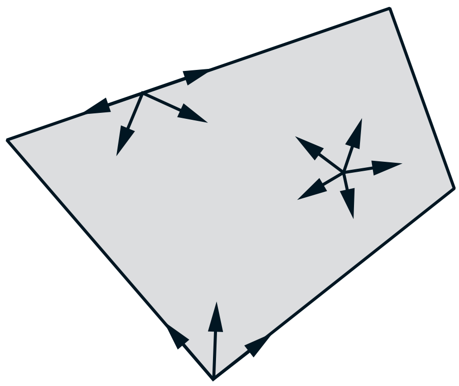
3 Optimality Conditions

3.1 Feasible directions

SLIDE 7

- We are at $\boldsymbol{x} \in P$ and we contemplate moving away from $\boldsymbol{x}$, in the direction of a vector $\boldsymbol{d} \in \Re^n$.
- We need to consider those choices of $\boldsymbol{d}$ that do not immediately take us outside the feasible set.
- A vector $\boldsymbol{d} \in \Re^n$ is said to be a **feasible direction** at $\boldsymbol{x}$, if there exists a positive scalar θ for which $\boldsymbol{x} + \theta\boldsymbol{d} \in P$.

SLIDE 8



SLIDE 9

- $\boldsymbol{x}$ be a BFS to the standard form problem corresponding to a basis $\boldsymbol{B}$.
- $x_i = 0,\ i \in N$, $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{B}$.
- We consider moving away from $\boldsymbol{x}$, to a new vector $\boldsymbol{x} + \theta\boldsymbol{d}$, by selecting a nonbasic variable x_j and increasing it to a positive value θ, while keeping the remaining nonbasic variables at zero.
- Algebraically, $d_j = 1$, and $d_i = 0$ for every nonbasic index i other than j.
- The vector $\boldsymbol{x}_B$ of basic variables changes to $\boldsymbol{x}_B + \theta\boldsymbol{d}_B$.
- Feasibility: $\boldsymbol{A}(\boldsymbol{x} + \theta\boldsymbol{d}) = \boldsymbol{B} \ \Rightarrow \ \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}$.
- $\boldsymbol{0} = \boldsymbol{A}\boldsymbol{d} = \sum_{i=1}^{n} \boldsymbol{A}_i d_i = \sum_{i=1}^{m} \boldsymbol{A}_{B(i)} d_{B(i)} + \boldsymbol{A}_j = \boldsymbol{B}\boldsymbol{d}_B + \boldsymbol{A}_j \Rightarrow \boldsymbol{d}_B = -\boldsymbol{B}^{-1}\boldsymbol{A}_j$.
- Nonnegativity constraints?
  - If $\boldsymbol{x}$ nondegenerate, $\boldsymbol{x}_B > \boldsymbol{0}$; thus $\boldsymbol{x}_B + \theta\boldsymbol{d}_B \geq \boldsymbol{0}$ for θ is sufficiently small.
  - If $\boldsymbol{x}$degenerate, then $\boldsymbol{d}$ is not always a feasible direction. Why?
- Effects in cost?
  Cost change: $\boldsymbol{c}'\boldsymbol{d} = c_j - \boldsymbol{c}'_B\boldsymbol{B}^{-1}\boldsymbol{A}_j$ This quantity is called **reduced cost** $\overline{c}_j$ of the variable x_j.

3.2 Theorem

SLIDE 10

- $\boldsymbol{x}$ BFS associated with basis B
- $\overline{\boldsymbol{c}}$ reduced costs
  Then
- If $\overline{\boldsymbol{c}} \geq \boldsymbol{0} \Rightarrow \boldsymbol{x}$ optimal
- $\boldsymbol{x}$ optimal and non-degenerate $\Rightarrow \overline{\boldsymbol{c}} \geq \boldsymbol{0}$

3.3 Proof

- $\boldsymbol{y}$ arbitrary feasible solution
- $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x} \Rightarrow \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b} \Rightarrow \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}$

SLIDE 11

$$\Rightarrow \boldsymbol{B}\boldsymbol{d}_B + \sum_{i \in N} \boldsymbol{A}_i d_i = \boldsymbol{0}$$

$$\Rightarrow \boldsymbol{d}_B = -\sum_{i \in N} \boldsymbol{B}^{-1}\boldsymbol{A}_i d_i$$

$$\begin{aligned} \Rightarrow \boldsymbol{c}'\boldsymbol{d} &= \boldsymbol{c}'_B\boldsymbol{d}_B + \sum_{i \in N} c_i d_i \\ &= \sum_{i \in N} (c_i - \boldsymbol{c}'_B\boldsymbol{B}^{-1}\boldsymbol{A}_i) d_i = \sum_{i \in N} \overline{c}_i d_i \end{aligned}$$

SLIDE 12

- Since $y_i \geq 0$ and $x_i = 0, i \in N$, then $d_i = y_i - x_i \geq 0, i \in N$
- $\boldsymbol{c}'\boldsymbol{d} = \boldsymbol{c}'(\boldsymbol{y} - \boldsymbol{x}) \geq 0 \ \Rightarrow \boldsymbol{c}'\boldsymbol{y} \geq \boldsymbol{c}'\boldsymbol{x}$

  $\Rightarrow \boldsymbol{x}$ optimal

  (b) Your turn

6.251J / 15.081J Introduction to Mathematical Programming
Fall 2009