

# Linear Programming

Guillaume Dalle (ENPC, CERMICS)

REOP - Class 5 (27/10/2021)

## Contents

<b>1 Homework</b>	<b>1</b>
1.1 Separation problem for the spanning tree polytope . . . . .	1
1.2 Reduction from 3-SAT to CLIQUE . . . . .	1
1.3 Dynamic programming for the knapsack problem . . . . .	2
<b>2 Modeling with linear functions</b>	<b>2</b>
2.1 Polyhedra . . . . .	2
2.2 Linear optimization problems . . . . .	2
2.3 Where to find an optimum? . . . . .	3
<b>3 Solution algorithms for LPs</b>	<b>3</b>
3.1 Simplex algorithm . . . . .	3
3.1.1 General idea . . . . .	3
3.1.2 Reduced costs . . . . .	3
3.2 Ellipsoid and interior point methods . . . . .	4
<b>4 Lagrangian duality</b>	<b>4</b>
<b>5 Advanced modeling examples</b>	<b>5</b>

## 1 Homework

### 1.1 Separation problem for the spanning tree polytope

For Q2, we assume a feasible solution  $x$  is given.

- Given a set of vertices  $X$ , we show that we can find a cut  $\varphi(X)$  with capacity  $|X| + \sum_{e \notin E(X)} x_e$ .
- Given a minimum cut  $B$  on the graph defined Q1, we show that we can find a cut  $B'$  of the same capacity and a set of vertices  $X$  such that  $B' = \varphi(X)$ .

### 1.2 Reduction from 3-SAT to CLIQUE

The 3-SAT problem can be described as follows:

- Input: A boolean formula  $F(x)$  on  $n$  binary variables  $x_1, \dots, x_n$  in conjunctive normal form (i.e. a conjunction  $\wedge = \text{AND}$  of  $\vee = \text{OR}$  clauses) with at most three literals by clause (a literal can be  $x_i$  or  $\neg x_i$ ).  
Example:  $F(x) = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$ .
- Output: **yes** if there is an  $x \in \{0, 1\}^n$  satisfying  $F$ , and **no** otherwise.

By the Cook-Levin theorem, 3-SAT is NP-complete. We show that this implies the maximum clique problem is NP-hard. Starting with an instance of 3-SAT, we can build an instance of CLIQUE using the graph shown on the Wooclap slide.

Since we didn't have time to correct this exercise during the session, if you want to see the solution, open this PDF book and go to page 1086.

### 1.3 Dynamic programming for the knapsack problem

The knapsack problem consists in filling a bag by choosing from a set of items  $i \in \{1, \dots, n\}$  with weights  $w_i$  and prices  $p_i$ . Its decision version asks whether there is a subset with total weight  $\leq W$  but total price  $\geq P$ .

We can build a dynamic programming algorithm to compute the value function

$$P(n, W) = \text{maximum price for a bag of weight } \leq W \text{ with the } n \text{ first items}$$

However its complexity depends on  $W$ , which makes it pseudo-polynomial. So this algorithm isn't sufficient to prove that KNAPSACK is in **P**.

## 2 Modeling with linear functions

### 2.1 Polyhedra

A polyhedron  $P$  is a finite intersection of closed half-spaces:

$$P = \bigcap_{i \in [m]} \{x \in \mathbb{R}^n : a_i x \leq b_i\} = \{x \in \mathbb{R}^n : Ax \leq b\}$$

Here the inequality  $Ax \leq b$  must be understood entrywise.

A polytope is a bounded polyhedron.

We say that  $x \in P$  is a vertex (or extreme point) of the polyhedron if  $x$  cannot be written as a non-trivial mean of two other points of  $P$ :

$$x = \frac{y+z}{2} \quad \text{with } y, z \in P \quad \implies \quad y = z = x$$

### 2.2 Linear optimization problems

A Linear Program is an optimization problem with linear objective and affine constraints:

$$\min_{x \in \mathbb{R}^n} c^\top x \quad \text{s.t.} \quad Ax \leq b \quad \iff \quad \min_{x \in P} c^\top x$$

In standard form, it is written

$$\min_x c^\top x \quad \text{s.t.} \quad Ax = b, x \geq 0 \tag{LP}$$

and we assume without loss of generality that  $\text{rank}(A) = m \leq n$  (no redundant constraints).

Exercise (Meunier - PL1): Prove that both forms above are equivalent.

A base  $B \subset [n]$  is a subset of  $m$  linearly independent columns of  $A$ : the submatrix  $A_B = (a_{i,j})_{i \in [m], j \in B}$  is nonsingular. Every base defines a basic solution of (LP)

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} \quad \text{with} \quad \begin{cases} x_B &= A_B^{-1}b \\ x_N &= 0 \end{cases}$$

which is called basic feasible if  $A_B^{-1}b \geq 0$ .

## 2.3 Where to find an optimum?

**Theorem** (Interpretation of bases):  $x$  is a basic feasible solution of (LP) iff it is a vertex of the polyhedron  $P$ .

*Proof:*

- Suppose  $x$  is a feasible basic solution. There is a basis  $B$  such that  $x_B = A_B^{-1}b \geq 0$  and  $x_N = 0$ . In addition, since  $Ay = A_B y_B = b$  we see that  $y_B = A_B^{-1}b = x_B$  and the same goes for  $z_B$ . Therefore,  $y = z = x$ .
- Now suppose that  $x$  is a vertex of  $P$ . Let  $K = \{j \in [n] : x_j > 0\}$ . Let  $d_K \in \ker A_K$  and  $d \in \mathbb{R}^n$  equal to  $d_K$  on  $K$  and equal to 0 elsewhere. Then for  $\epsilon > 0$  small enough,  $x \pm \epsilon d \in P$ , which means that  $d = 0$ . Thus  $A_K$  has independent columns, and since  $\text{rank} A = m$ ,  $K$  can be completed into a set  $B$  of size  $m$  such that  $A_B$  is invertible. As a consequence,  $B$  is a basis and  $x = (x_B, x_N)$  with  $x_N = 0$  and  $x_B = A_B^{-1}b \geq 0$ .

**Theorem** (Existence of an optimum): If  $P \neq \emptyset$  and  $\inf_{x \in P} c^\top x > -\infty$ , then (LP) has an optimal solution.

*Proof* (for the bounded case):

If  $P$  is a polytope,  $P$  is compact and the result follows.

**Theorem 9.1** (Optimal basis): If (LP) has an optimal solution, then at least one of the optimal solutions is basic (feasible).

*Proof* (for the bounded case):

We admit the Minkowski-Weyl theorem (every polytope is the convex hull of its vertices). Let  $x$  be an optimal solution of (LP), then  $x = \sum_k \lambda_k v_k$  where the  $v_k$  are the vertices of  $P$  and the  $\lambda_k \geq 0$  sum to 1. By concavity of  $x \mapsto c^\top x$ , we know that  $c^\top x \geq \sum_k \lambda_k (c^\top v_k)$ , hence there is a  $k$  such that  $c^\top x \geq c^\top v_k$ . Then  $v_k$  is an optimal solution, and it is basic feasible.

## 3 Solution algorithms for LPs

Naive algorithm: enumerate all bases, discard the infeasible ones and compare the feasible ones. Unfortunately, this method has exponential runtime.

### 3.1 Simplex algorithm

#### 3.1.1 General idea

Starting from a given vertex, try to find one of its neighbors with smaller cost. The way we choose the neighbor is called a pivot rule.

To find the initial vertex, we run a first simplex on an auxiliary problem which is easier to initialize.

#### 3.1.2 Reduced costs

Going from one vertex to the next is like going from one basis to the next. We can rewrite the constraints of (LP) by decomposing  $x$  along the current basis  $B$ :

$$Ax = b \iff A_B x_B + A_N x_N = b \iff x_B = A_B^{-1}(b - A_N x_N)$$

Starting from  $B$ , we rewrite the objective as:

$$\begin{aligned}
c^\top x &= c_B^\top x_B + c_N^\top x_N \\
&= c_B^\top (A_B^{-1}(b - A_N x_N)) + c_N^\top x_N \\
&= \underbrace{c_B^\top A_B^{-1} b}_{\text{cost of the basic solution}} + \underbrace{(c_N^\top - A_N^\top (A_B^{-1})^\top c_B^\top)^\top}_{\text{reduced cost } r_N^\top} x_N \\
&= \text{cost}(\text{sol}(B)) + r_N^\top x_N
\end{aligned}$$

**Proposition 9.2** (Role of the reduced cost): If  $r_N \geq 0$ , then  $B$  is an optimal basis. Otherwise, we can decrease the objective value by increasing  $x_j$  where  $j \in N$  is such that  $r_j < 0$ . This either leads to a new basis  $B' \subset B \cup \{j\}$  or to a direction of unboundedness.

Exercise (Meunier - PL11): Basic principle of column generation

Let  $(P)$  be the linear program

$$\min c^\top x \quad \text{s.t.} \quad Ax = b, x \in \mathbb{R}_+^n$$

We suppose that  $n$  is very large. Let  $I \subset [n]$ , we consider the program  $(P^I)$ :

$$\min c_I^\top x' \quad \text{s.t.} \quad Ax' = b, x' \in \mathbb{R}_+^I$$

We denote by  $(D)$  and  $(D^I)$  the respective dual problems of  $(P)$  and  $(P^I)$ . Let  $\tilde{y}$  be an optimal solution of  $(D^I)$ . Show that if  $\tilde{y}$  is also a feasible solution to  $(D)$ , then the optimal value of  $(P^I)$  is the same as the optimal value of  $(P)$ .

Exercise 9.5: Dual simplex

## 3.2 Ellipsoid and interior point methods

The simplex is efficient in practice, but has exponential worst-case running time for all known pivot rules. Other methods, discovered much later, have a much better theoretical complexity (polynomial even in the worst case): ellipsoid algorithm, interior point algorithms.

In particular, the ellipsoid algorithm runs in polynomial time even with an exponential number of constraints, as long as the separation problem can be solved efficiently.

## 4 Lagrangian duality

**Proposition 9.4** (Expression of the dual): The dual of the Linear Program in standard form (LP) is:

$$\max_{y \in \mathbb{R}^m} b^\top y \quad \text{s.t.} \quad A^\top y \leq c \quad (\text{LPD})$$

Exercise: Prove proposition 9.4

*Proof:* The Lagrangian for (LP) writes:

$$\mathcal{L}(x, \lambda, \mu) = c^\top x + \lambda^\top (b - Ax) + \mu^\top (-x)$$

where  $\lambda \in \mathbb{R}^m$  and  $\mu \in \mathbb{R}_+^n$  are Lagrange multipliers associated with each constraint. The dual problem is defined as

$$\max_{\lambda, \mu \geq 0} \underbrace{\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \mu)}_{\text{dual function } g(\lambda, \mu)}$$

Exercise: Prove proposition 6.12.

**Theorem 8.1** (Weak duality): We always have  $\text{val}(LPD) \leq \text{val}(LP)$ .

**Theorem 9.5** (Strong duality in Linear Programming): If either (LP) or (LPD) has a feasible solution, then the values of both problems are equal.

*Proof:* Based on the termination of the simplex algorithm and a correspondence between the final reduced costs & the dual variables.

**Proposition 9.6** (Complementary slackness): Let  $x$  and  $y$  be feasible solutions of the primal and the dual respectively. Then  $x$  and  $y$  are both optimal solutions iff

$$(c - A^T y)^T x = 0 \quad \text{and} \quad y^T (b - Ax) = 0$$

*Proof:* Follows directly from weak duality.

Remark: The dual variables have an economic interpretation. In the standard form LP, we replace the constraint  $Ax = b$  by  $Ax = b + \varepsilon$ , then the value of the problem will change by approximately  $-(y^*)^T \varepsilon$ , where  $y^*$  is the optimal dual solution.

## 5 Advanced modeling examples

Exercise (Meunier - PL6): Computing sums of order statistics

For any vector  $x \in \mathbb{R}^n$ , we denote by  $x^{[i]}$  the  $i$ -th largest element of the set  $\{x_1, \dots, x_n\}$ . For instance,  $x^{[1]}$  is the largest component of  $x$ , and  $x^{[n]}$  is the smallest.

1. Let  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$ . We fix an integer  $k \in [n]$ . Show that the problem

$$\min \sum_{i=1}^k x^{[i]} \quad \text{s.t.} \quad Ax = b, x \in \mathbb{R}_+^n$$

can be formulate as a Linear Program. Hint: introduce one constraint per subset  $S \subset [n]$  of size  $k$ .

2. (Harder) Show that there is a formulation with at most  $m + n$  constraints.

Exercise (Meunier - course notes): Nash equilibria in zero-sum games