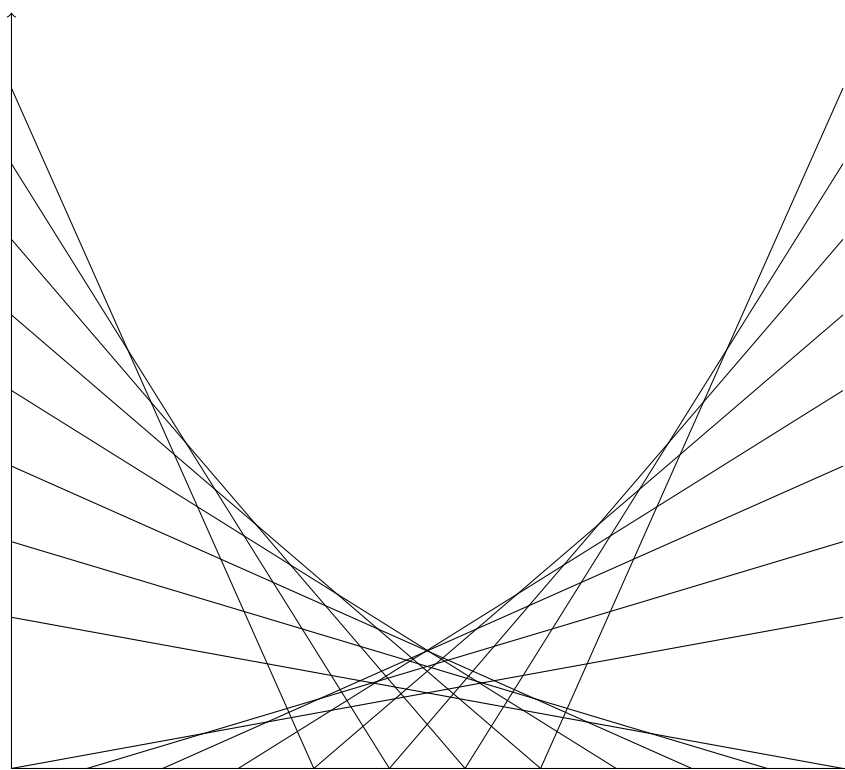


---

# Optimization Handbook

---



Henri Lefebvre

May 28, 2019

# Contents

<b>1</b>	<b>Introduction to Optimization</b>	<b>4</b>
<b>I</b>	<b>Linear Optimization</b>	<b>5</b>
<b>2</b>	<b>Introduction to linear programming</b>	<b>6</b>
1	Problems description . . . . .	6
1.1	Definitions . . . . .	6
1.2	Standard form . . . . .	7
2	Fundamental Theorem of LP . . . . .	7
3	Examples . . . . .	9
3.1	Minimum cost flow problem . . . . .	9
3.2	Support Vector Machine . . . . .	9
3.3	Knapsack problem . . . . .	9
<b>3</b>	<b>The Simplex algorithm</b>	<b>10</b>
1	Formal derivation . . . . .	10
1.1	Assumptions . . . . .	10
1.2	Vector leaving the basis . . . . .	11
1.3	Vector entering the basis . . . . .	11
2	Geometrical Interpretation . . . . .	11
3	Finding a feasible solution . . . . .	11
4	The Simplex method . . . . .	11
4.1	Pseudo-code . . . . .	11
4.2	Degeneracy . . . . .	11
4.3	Examples . . . . .	11
5	The revised Simplex . . . . .	11
5.1	Matrix form of the Simplex . . . . .	11
5.2	Pseufo-code . . . . .	11
5.3	Examples . . . . .	11
6	The bounded Simplex . . . . .	11
6.1	Formal derivation . . . . .	11
6.2	Pseudo-code . . . . .	11
6.3	Examples . . . . .	11

7	The transportation Simplex . . . . .	11
7.1	Formal derivation . . . . .	11
7.2	Pseudo-code . . . . .	11
7.3	Examples . . . . .	11
<b>4</b>	<b>Lagrangian duality</b>	<b>12</b>
1	Motivations . . . . .	12
2	Practical derivations . . . . .	12
3	Duality theorems . . . . .	12
4	Geometric interpretation . . . . .	12
5	Sensitivity . . . . .	12
6	Complementary slackness . . . . .	12
7	The Primal-Dual algorithm . . . . .	12
7.1	Examples . . . . .	12
<b>5</b>	<b>The branch-and-bound algorithm</b>	<b>13</b>
<b>6</b>	<b>The branch-and-cut algorithm</b>	<b>14</b>
<b>7</b>	<b>Column generation and the branch-and-price algorithm</b>	<b>15</b>
<b>8</b>	<b>Relaxation techniques</b>	<b>16</b>
1	Formal definition . . . . .	16
2	Linear relaxation . . . . .	16
3	Lagrangian relaxation . . . . .	16
4	Surrogate relaxation . . . . .	16
<b>9</b>	<b>Benders decomposition</b>	<b>17</b>
1	Introduction . . . . .	17
2	Formal derivation . . . . .	17
3	Algorithm . . . . .	19
3.1	Pseudo code . . . . .	19
3.2	Generating a feasibility cut . . . . .	19
4	Stabilisation methods . . . . .	19
4.1	Bundle methods . . . . .	19
4.2	Proximal methods . . . . .	19
5	Generalization . . . . .	19
<b>10</b>	<b>Dantzig-Wolfe decomposition</b>	<b>21</b>
<b>11</b>	<b>Robust optimization</b>	<b>22</b>
<b>12</b>	<b>Stochastic optimization</b>	<b>23</b>

<b>II</b>	<b>Non-linear Optimization</b>	<b>24</b>
<b>13</b>	<b>Introduction</b>	<b>25</b>
<b>14</b>	<b>Descent methods</b>	<b>26</b>
<b>15</b>	<b>Newton methods</b>	<b>27</b>
<b>16</b>	<b>Non-differentiable optimization</b>	<b>28</b>
	<b>Appendices</b>	<b>29</b>
<b>A</b>	<b>Convex sets</b>	<b>30</b>
1	Polyhedra . . . . .	30
1.1	Definitions . . . . .	30
1.2	Theorems . . . . .	30

## Chapter 1

# Introduction to Optimization

[2]

**Part I**

**Linear Optimization**

# Chapter 2

## Introduction to linear programming

### 1 Problems description

#### 1.1 Definitions

Linear programming (LP) takes interests in problems in which both the objective function and the constraints are linear with respect to the decision variables. The feasible region is therefore composed of linear inequalities or linear equalities. For the sake of generality, we often consider a problem of minimizing a linear objective function subject to equality constraints. Such a problem is said to be in *standard form*. Note that the standard form is only used as a way to present a homogeneous framework for theoretical work. In the following section, we show how any formulated linear program can be reduced to a problem written in standard form.

Some examples of problems which can be formulated in a linear fashion are presented in section 3. Yet, we give here a very simple example for the sake of understanding. This problem will also allow us to have a first geometrical interpretation of some mathematical objects considered in linear programming.

Let us consider a production plant where two kinds of items are to be produced. Both items are made of two raw materials  $RM_1$  and  $RM_2$ . Items of type  $A$  need 10 units of  $RM_1$  and 4 units of  $RM_2$  while items of type  $B$  need 2 units of  $RM_1$  and 4 units of  $RM_2$  to be produced. The problem is to decide how many items of type  $A$  and of type  $B$  should be produced in the plant, knowing that the raw materials are of

fixed amount. Let us assume that we have 50 units of raw material  $RM_1$  and 60 units of  $RM_2$ . A so-called *feasible solution* is a decision which satisfies the capacity constraints of the raw materials (i.e., we do not produce more items than we have raw material to do so). A feasible solution is said to be *optimal* if it minimizes or maximizes a certain quantity. Here, we will consider the minimization of the cost. Let us assume that items of type  $A$  cost 5 euros to be produced and that items  $B$  cost 4 euros. The problem can be stated as :

$$\begin{aligned} &\text{minimize } 3x + 4y \\ &\text{s.t. } 10x + 4y \leq 50 \\ &\quad 2x + 3y \leq 30 \\ &\quad x \geq 0, y \geq 0 \end{aligned}$$

Here, one may argue that the decisions variables  $x$  and  $y$  need to take integer values. This in fact depends on the context and application of the problem. We will see however that integer linear programming (ILP) problems as well as mixed integer linear programming (MILP) where we have both continuous and integer decision variables are much harder to solve. Yet we will consider quite efficient algorithms to solve such kinds of problems, among which : the branch-and-bound approach and the cutting planes algorithm.

Since our problem is a two-dimensional problem, in that sense that we have two decision real (or integer) variables to decide, one can plot the *feasible region* of the problem. The feasible region denotes the set

of feasible solutions. Figure 2.1 depicts the feasible region of our problem.

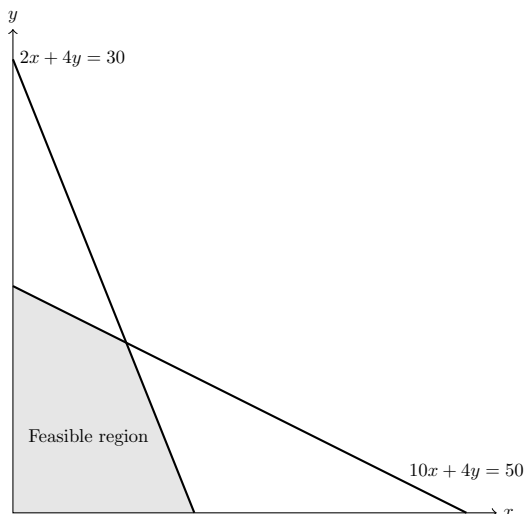


Figure 2.1: 2D representation of a feasible region

One should indeed see that above this feasible region stands a plane (or hyperplane in higher dimensions) defining the objective function. Since the objective function is a plane, it is clear that the optimal solution can be found in one of the extreme points of the feasible region. This property, in fact, holds in general and is called the *Fundamental theorem of linear programming* and will be formally introduced in section 2.

## 1.2 Standard form

### Formal definition

**Definition 1** (Standard form). *A linear programming problem is said to be in standard form if it is written as*

$$\begin{aligned} & \text{minimize } c^T x \\ & \text{s.t. } Ax = b \\ & \quad x \geq 0 \end{aligned}$$

### Examples

In this section, we show how we can turn linear problems which are not originally in the standard form to a problem written in standard form.

**Negative variables** If a given problem uses a negative variable, say  $x \leq 0$ . It suffices to consider the opposite decision variable  $\hat{x} = -x$ . The problem is now in standard form.

**Real variables** If a given problem uses a free variable (i.e., a variable which can be positive or negative)  $x \in \mathbb{R}$ . We introduce two positive decision variables  $x^+ \geq 0$  and  $x^- \geq 0$  and write  $x$  as  $x^+ - x^- \in \mathbb{R}$ . The problem is now in standard form.

**Inequality constraints** If a given problem defines the feasible region with inequality constraint, so-called *slack* variables can be introduced. Considering an inequality constraint  $\sum_j a_j x_j \leq b$ , we introduce variable  $s \geq 0$  so that  $\sum_j a_j x_j + s = b$ . We do not restrict the values of  $s$  (other than by its sign) and do not associate any cost in the objective function. Therefore, the value of  $s$  in the optimal solution corresponds to the  $b - \sum_j a_j x_j$ , hence the name of slack variables. The obtained problem is now in standard form.

## 2 Fundamental Theorem of LP

The fundamental theorem of linear programming has been intuitively introduced in the previous section. We enounce it here formally and prove its validity and geometrical interpretation. First, we introduce the following assumption which will hold in general in the subsequent theorems :

**Assumption 1** (Full rank assumption). *Considering the feasible region  $\{x | Ax = b, x \geq 0\}$  where  $A$  is a  $m \times n$  matrix. We assume that  $\text{rank}(A) = n$*



The fundamental theorem of LP characterizes the optimal solutions of a given problem. In that sense, it reduces the search space for optimality. To properly enounce the theorem, we need to introduce the concept of *basic* solutions.

Consider a feasible region defined as

$$\{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$$

where  $A$  is a  $m \times n$ -matrix of full rank (see assumption 1). If we select  $n$  linearly independent columns from  $A$ , then the set of columns represent a basis of  $\mathbb{R}^n$ . Let us denote by  $B$  the matrix corresponding to that basis. Since  $B$  is non-singular, the following system can be solved uniquely :

$$Bx_B = b$$

where  $x_B$  is an  $m$ -dimensional vector. Then, clearly, the vector  $x$  defined as  $x = [x_B, \mathbf{0}]$  belongs to the feasible region since  $Ax = A[x_B, \mathbf{0}] = Ax_B + A\mathbf{0} = b$ . This consideration leads to the following definition :

**Definition 2** (Basic solution). *Considering a feasible region  $\{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$  where  $A$  is a matrix of full rank. A vector  $x$  is said to be basic if and only if there exists a basis  $B$  of  $\mathbb{R}^n$  composed of  $n$  columns of  $A$  and  $x = [B^{-1}b, \mathbf{0}]$ . If, moreover, it is positive element-wise then it is said to be a feasible basic solution.*

We can now enounce the theorem :

**Theorem 1** (Fundamental theorem of linear programming). *Consider the following LP problem in standard form :*

$$\begin{aligned} & \text{minimize } c^T x \\ & \text{s.t. } Ax = b \\ & \quad x \geq 0 \end{aligned}$$

Then, under the full rank assumption (1),

- (i) if there is a feasible solution, there is a basic feasible solution
- (ii) if there is an optimal feasible solution, there is an optimal basic feasible solution.

*Proof.*

(i)  $\Rightarrow$  :

$\Leftarrow$  :

(ii)  $\Rightarrow$  :

$\Leftarrow$  :

□

**Theorem 2** (Equivalence between basic solutions and extreme points). *Considering a matrix  $A$  of full rank (see assumption 1), a vector  $x$  is an extreme point of the polyhedron  $\{x | Ax = b, x \geq 0\}$  if and only if it is a basic feasible solution.*

*Proof.*

(i)  $\Rightarrow$  :

$\Leftarrow$  :

(ii)  $\Rightarrow$  :

$\Leftarrow$  :

□

One final remark should be made regarding the definition of a basic solution. Indeed, in general, the coefficients of a basic solution may not be all non-zero. We therefore give the following definition :

**Definition 3** (Degenerate solution). *A basic solution is said to be degenerate if and only if one or more of the basic variables value are zero.*

Note that, in presence of degeneracy, ambiguity arises since one could interchange freely the zero-valuated basic variables with non-basic variables.

### **3 Examples**

**3.1 Minimum cost flow problem**

**3.2 Support Vector Machine**

**3.3 Knapsack problem**

## Chapter 3

# The Simplex algorithm

### 1 Formal derivation

#### 1.1 Assumptions

**Assumption 2** (Nondegeneracy assumption). *Every basic feasible solution is a nondegenerate basic feasible solution.*

- 1.2 Vector leaving the basis
- 1.3 Vector entering the basis
- 2 Geometrical Interpretation
- 3 Finding a feasible solution
- 4 The Simplex method
  - 4.1 Pseudo-code
  - 4.2 Degeneracy
  - 4.3 Examples
- Optimal solution
- Degenerate solution
- Unbounded problem
- Infeasible problem
- 5 The revised Simplex
  - 5.1 Matrix form of the Simplex
  - 5.2 Pseufo-code
  - 5.3 Examples
- 6 The bounded Simplex
  - 6.1 Formal derivation
  - 6.2 Pseudo-code
  - 6.3 Examples
- 7 The transportation Simplex
  - 7.1 Formal derivation
  - 7.2 Pseudo-code
  - 7.3 Examples

## Chapter 4

# Lagrangian duality

- 1 Motivations
- 2 Practical derivations
- 3 Duality theorems
- 4 Geometric interpretation
- 5 Sensitivity
- 6 Complementary slackness
- 7 The Primal-Dual algorithm
  - 7.1 Examples

## Chapter 5

# The branch-and-bound algorithm

## Chapter 6

# The branch-and-cut algorithm

## Chapter 7

# Column generation and the branch-and-price algorithm



## Chapter 8

# Relaxation techniques

- 1 Formal definition
- 2 Linear relaxation
- 3 Lagrangian relaxation
- 4 Surrogate relaxation

# Chapter 9

## Benders decomposition

### 1 Introduction

Benders decomposition is a solution method for solving certain large-scale optimization problems. It is particularly suited for problems in which a set of variables are said to be *complicating* in the sense that fixing them to a given value makes the problem easy. Briefly, the Benders decomposition approach separates an original problem into several decision stages. A first-stage *master* problem is solved using only a subset of variables, then, the values of the remaining variables are determined by a so-called *subproblem* depending on the first-stage variables. If the master problem's optimal solution yields an infeasible subproblem, a *feasibility cut* is added to the master problem, which is then re-solved. Due to the structure of the reformulation, the Benders algorithm starts with a *restricted master problem* where only a subset of constraints are considered while the others are iteratively added.

This technique was first introduced in [1] and has since been generalized to non-linear mixed integer problems.

### 2 Formal derivation

Consider the following problem :

$$\text{minimize } c^T x + f(y) \quad (9.1)$$

$$\text{s.t. } Ax + g(y) = b \quad (9.2)$$

$$y \in Y, x \geq 0 \quad (9.3)$$

where variable  $y$  is a *complicating constraint*. Note that it may be complicating due to the form of  $f$  or  $g$  but also by our ability to enforce the constraint  $y \in Y$ . We assume that fixing  $y$  to a given value  $\hat{y}$  turns our problem into an easy-to-solve problem.

We can notice that our problem is equivalent to the following one :

$$\min \{ f(y) + \min \{ c^T x : Ax = b - g(y) \} : y \in Y \}$$

Let us denote by  $q(y)$  the value of the minimization problem over  $x$  :  $q(y) = \min \{ c^T x : Ax = b - g(y) \}$ . By duality, the following holds

$$q(y) = \max \{ (b - g(y))^T \pi : A^T \pi \leq c \}$$

Note that the feasibility space of the dual does not depend on the values of  $y$  and let us apply the decomposition theorem for polyhedra 5 on it :

$$\{ A^T \pi \leq c \} = \left\{ \sum_i u^i \alpha_i + \sum_j v^j \beta_j \right\}$$

$$\sum_i \alpha_i = 1 \quad \text{and} \quad \alpha, \beta \geq 0$$

where  $\{u^i\}_i$  denotes the extreme points of  $\{x | Ax \leq c\}$  and  $\{v^j\}_j$  denotes the extreme rays of the polyhedral cone  $\{x | Ax \leq 0\}$ . Intuitively, the convex combination of the extreme points of  $\{x | Ax \leq c\}$  defines the optimal solutions (since we know that there exists at

least one optimal solution corresponding to an extreme point of the considered polyhedron) while the conical combination of extreme rays of  $\{x|Ax \leq 0\}$  defines the feasibility region. The following theorem will allow us to reformulate our problem :

**Theorem 3.** *Let  $(\mathcal{P})$  be the following problem :*

$$\max\{c^T x : Ax \leq b, x \geq 0\} \quad (\mathcal{P})$$

*Then,  $(\mathcal{P})$  is upper bounded if and only if*

$$c^T v^j \leq 0 \quad \forall j = 1, \dots, J$$

*where  $\{v^j\}_{j=1, \dots, J}$  denotes the set of extreme rays of  $\{x|Ax \leq 0\}$ .*

*Proof.*  $\Rightarrow$  : By contradiction, let us suppose that  $(\mathcal{P})$  is upper bounded and that there exists  $k$  such that  $c^T v^k > 0$ . Let us consider a feasible solution to  $(\mathcal{P})$  denoted by  $u = z + t$  where  $z$  is a convex combination of the extreme points of  $\{x|Ax \leq b\}$  and  $t$  an element of the conical combinations of the extreme rays of  $\{x|Ax \leq 0\}$  (see theorem 5). Let  $\lambda \in \mathbb{R}_+$ . Since  $v^k$  is in the conical polyhedra  $\{x|Ax \leq 0\}$ , it holds that  $\lambda A v^k \leq 0$ . Moreover, we have  $A t \leq 0$ . Hence,  $A(t + \lambda v^k) \leq 0$ . Now, since  $v^k \geq 0$ , we have  $\lambda v^k \geq 0$  and since  $t \geq 0$  it holds that  $t + \lambda v^k \geq 0$ . This shows that, for any  $\lambda$ ,  $z + \lambda v^k$  is a feasible solution for problem  $(\mathcal{P})$ . However, its associated objective value is given by  $c^T u + \lambda c^T v^k \rightarrow +\infty$  when  $\lambda \rightarrow +\infty$  since, by assumption,  $c^T v^k > 0$ . This contradicts the fact that  $(\mathcal{P})$  is upper bounded.

$\Leftarrow$  : Let us consider a solution  $u = z + t$  then  $c^T u = c^T z + \sum_j c^T v^j \beta_j \leq c^T z$  since  $c^T v^j \leq 0, \forall j = 1, \dots, J$ . Problem  $(\mathcal{P})$  is therefore upper bounded by  $\sum_i \alpha_i \max_i \{c^T u^i\}$ .  $\square$

Theorem 3 can be intuitively understood by interpreting  $c^T v^j$  as  $\langle c, v^j \rangle$  (scalar product). The necessary and sufficient condition that  $\langle c, v^j \rangle \leq 0$  simply expresses that the hyperplane defining the objective function must be *oriented* (i.e., increasing) in the direction of the origin of the cone formed by the extreme rays of the polyhedron. It is depicted in figure 9.1.

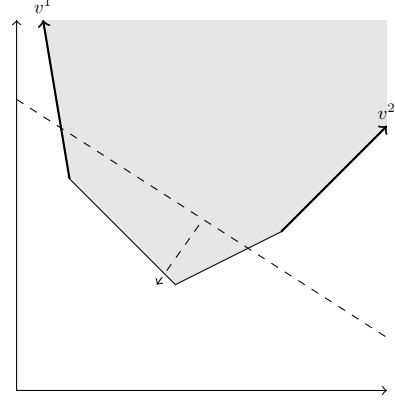


Figure 9.1: Illustration of theorem 3

If we apply theorem 3 to our specific problem, we find that a necessary and sufficient condition for  $q(y)$  to be upper bounded is that  $(b - g(y))^T v^j \leq 0, \forall j = 1, \dots, J$  where  $v^j$  denotes an extreme ray of the polyhedron  $\{x|Ax \leq c\}$  and  $J$  a list for their indices. Moreover, we know that there exists at least one optimal solution realised in an extreme point of the feasible region. Therefore, by calling  $u^i, i = 1, \dots, I$  the extreme points of the feasible region, the problem of finding the value of  $q(y)$  can be reformulated as

$$q(y) = \min\{q : q \geq (b - g(y))^T u^i \quad \forall i = 1, \dots, I\}$$

Assuming that this problem is bounded. We finally can write our original problem as the following

$$\text{minimize } f(y) + q \quad (9.4)$$

$$\text{s.t. } y \in Y \quad (9.5)$$

$$(b - g(y))^T u^i \leq q \quad \forall i = 1, \dots, I \quad (9.6)$$

$$(b - g(y))^T v^j \leq 0 \quad \forall j = 1, \dots, J \quad (9.7)$$

where constraints 9.6 are called *optimality cuts/constraints* since they define the extreme points of the feasible region and constraints 9.7 are called *feasibility cuts/constraints* since they enforce that the problem is bounded.

### 3 Algorithm

#### 3.1 Pseudo code

It is clear that our final formulation implies an exponential number of constraints since polyhedra typically have an exponential number of extreme points and extreme rays. The idea of the Benders Decomposition Algorithm is to work with a relaxation of the problem where only a limited number of constraints are considered. The algorithm then tries to reach the optimality of the original problem by a *clever* choice of constraints to be added iteratively. The algorithm is presented in 1.

#### 3.2 Generating a feasibility cut

In **Step 2** of algorithm 1, it is asked to find an extreme ray such that  $(b - g(y))^T v^j > 0$ , i.e., a direction in which the problem is unbounded. A way to do that is to use the Simplex Tableau. Indeed, if a problem is unbounded, this implies that there exists a variable whose reduced cost is positive (i.e., ready to enter the basis) while the associated column is composed of positive terms (i.e., no constraint bounds its value). The associated column is in fact an extreme ray of the polyhedron.

### 4 Stabilisation methods

#### 4.1 Bundle methods

#### 4.2 Proximal methods

### 5 Generalization

In this section, we take interest in a generalization of the Benders decomposition applicable for the following problem

$$\text{minimize } f(x, y) \quad (9.8)$$

$$\text{s.t. } g(x, y) \leq 0 \quad (9.9)$$

---

#### Algorithm 1 Benders Decomposition Algorithm

---

**Step 0** : Find an extreme point of  $\{x | Ax \leq c, x \geq 0\}$  (e.g., via the phase 1 of the Simplex)  
 $1 \rightarrow p, 0 \rightarrow k$

**Step 1** : Solve relaxed problem with only  $p$  optimality constraints and  $k$  feasibility constraints :

$$\begin{aligned} &\text{minimize } f(y) + q \\ &\text{s.t. } y \in Y \\ &\quad (b - g(y))^T u^i \leq q \quad \forall i = 1, \dots, p \\ &\quad (b - g(y))^T v^j \leq 0 \quad \forall j = 1, \dots, k \end{aligned}$$

Let  $\bar{y}, \bar{q}$  be the optimal solution thus obtained.

**Step 2** : Check the feasibility of  $(\bar{y}, \bar{q})$  for the original problem by solving

$$q(\bar{y}) = \max\{(b - g(y))^T \pi : A^T \pi \leq c, \pi \geq 0\}$$

Then,

**If**  $q(\bar{y}) = +\infty$  :

The problem is unbounded and, from theorem 3, one can find an extreme ray  $v^j$  such that  $(b - g(y))^T v^j > 0$ .

Add feasibility cut to the relaxed problem.

Increment  $k$ .

Got to **Step 1**.

**Else** :

Let  $\bar{\pi}$  be the optimal solution of cost  $q(\bar{y})$ .

**If**  $\bar{q} < q(\bar{y})$  :

Add optimality cut  $(b - g(y))^T \bar{\pi} \leq q$

Increment  $p$ .

Got to **Step 1**.

**Else** :

The solution is optimal.

---

$$x \in X \quad (9.10)$$

$$y \in Y \quad (9.11)$$

under the following hypothesis :

(i)  $X$  is a convex set

(ii)  $f(x, y)$  and  $g(x, y)$  are convex-in- $x$  over  $X$

Again, we can write the *projection* on the  $y$ -variable space, thus obtaining :

$$\min \{ \inf \{ f(x, y) : g(x, y) \leq 0, x \in X \} : y \in Y \}$$

and let us denote by  $h(y)$  the minimization problem over  $x$ , i.e.,

$$h(y) = \inf \{ f(x, y) : g(x, y) \leq 0, x \in X \}$$

## Chapter 10

# Dantzig-Wolfe decomposition

## Chapter 11

# Robust optimization

## Chapter 12

# Stochastic optimization



## Part II

# Non-linear Optimization

## Chapter 13

# Introduction

## Chapter 14

# Descent methods

## Chapter 15

# Newton methods

## Chapter 16

# Non-differentiable optimization

# Appendices

# Appendix A

## Convex sets

### 1 Polyhedra

#### 1.1 Definitions

##### Combinations and hulls

First, the following definitions will allow us to introduce some geometrical notions :

**Definition 4** (Convex combination). *Let  $x_1, \dots, x_n$  be a finite set of vectors in a real vector space, a convex combination of these vectors is a vector of the form*

$$\sum_{i=1}^n \alpha_i x_i \quad \text{with} \quad \sum_{i=1}^n \alpha_i = 1, \alpha_i \geq 0$$

**Definition 5** (Convex hull).

$$\text{conv}(X) = \left\{ \sum_{i=1}^n \alpha_i x_i \mid x_i \in X, \sum_{i=1}^n \alpha_i = 1, \alpha_i \geq 0 \right\}$$

**Definition 6** (Conical combination). *Let  $x_1, \dots, x_n$  be a finite set of vectors in a real vector space, a conical combination of these vectors is a vector of the form*

$$\sum_{i=1}^n \alpha_i x_i \quad \text{with} \quad \alpha_i \geq 0$$

**Definition 7** (Conical hull).

$$\text{cone}(X) = \left\{ \sum_{i=1}^n \alpha_i x_i \mid x_i \in X, \alpha_i \geq 0 \right\}$$

#### Polyhedra, polytopes and polyhedral cones

**Definition 8** (Polyhedron). *(plural : polyhedra)*

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$$

**Definition 9** (Polytope). *A polytope is the convex set a finite number of points. Alternatively,*

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b, u^- \leq x \leq u^+\}$$

**Definition 10** (Polyhedral cone).

$$P = \{x \in \mathbb{R}^n \mid Ax \leq 0, x \geq 0\}$$

**Observation 1** (Important). *Note that no real consensus has been reached on the definitions of polyhedra and polytopes. Sometimes, polyhedra corresponds to three dimensional solids with polygonal faces while polytopes denote the extension of a polyhedra to higher dimensions.*

It comes from these definitions that every polytopes and every polyhedral cones are polyhedra. These geometrical objects are depicted in figure A.1.

#### 1.2 Theorems

We first recall the following definition in order to properly enounce some important theorems.

**Definition 11** (Minkowski sum). *Let  $A$  and  $B$  be two vector spaces, the Minkowski sum is defined as*

$$A + B = \{a + b \mid a \in A, b \in B\}$$

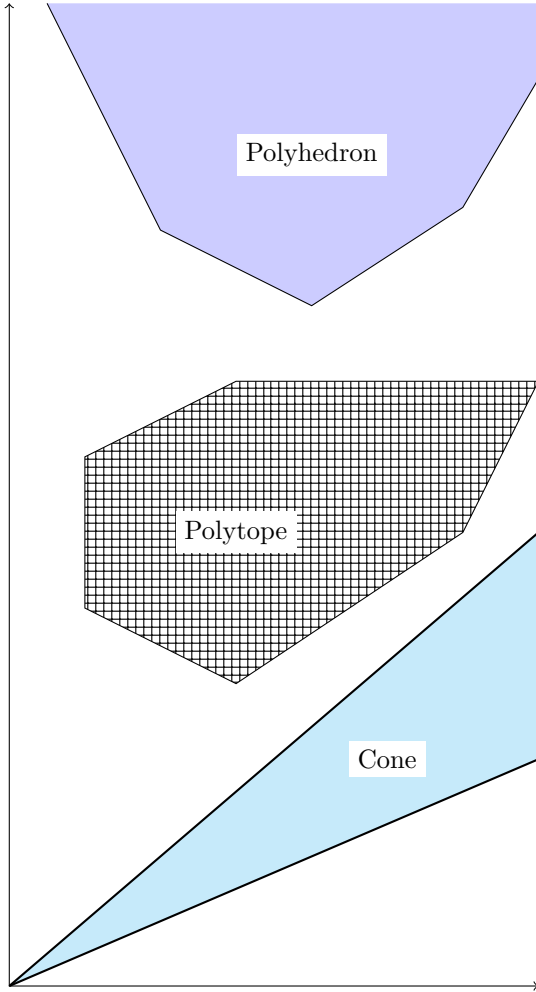


Figure A.1: A polyhedron, a polytope and a polyhedral cone depicted in 2D

**Theorem 4** (Affine Minkowski-Weyl). *Let there be a polyhedron defined by a set of inequalities,  $P = \{x \in \mathbb{R}^n | Ax \leq b\}$ . There exists vectors  $x_1, \dots, x_q \in \mathbb{R}^n$  and  $y_1, \dots, y_r \in \mathbb{R}^n$  such that*

$$P = \text{cone}(x_1, \dots, x_q) + \text{conv}(y_1, \dots, y_r)$$

**Theorem 5** (Decomposition theorem for polyhedra). *A set  $P$  of vectors in a Euclidean space is a polyhedron if and only if it is the Minkowski sum of a poly-*

*tope  $Q$  and a polyhedral cone  $C$ .*

$$P = Q + C$$

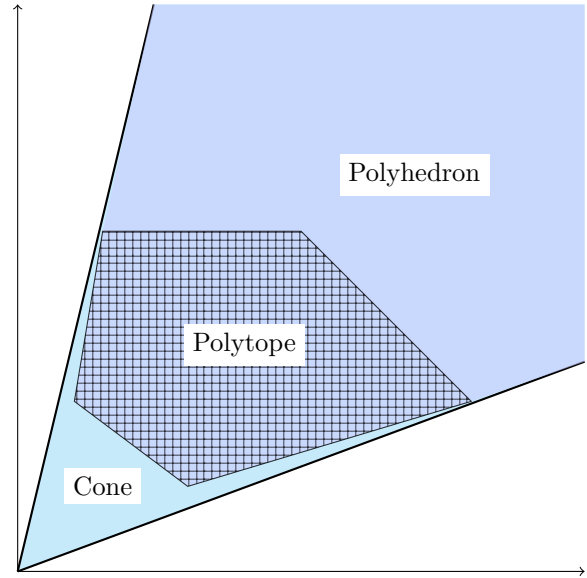


Figure A.2: Illustration of theorem 5

This decomposition theorem is illustrated in figure A.2. Intuitively, the polytope defines the *lower* shape of the polyhedron and the polyhedral cone defines the *queue* of the polyhedron.



# Bibliography

- [1] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, Dec 1962.
- [2] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, 1973.

# List of Figures

2.1	2D representation of a feasible region . . . . .	7
9.1	Illustration of theorem 3 . . . . .	18
A.1	A polyhedron, a polytope and a polyhedral cone depicted in 2D . . . . .	31
A.2	Illustration of theorem 5 . . . . .	31

# List of Tables

# List of Algorithms

1	Benders Decomposition Algorithm . . .	19
---	---------------------------------------	----