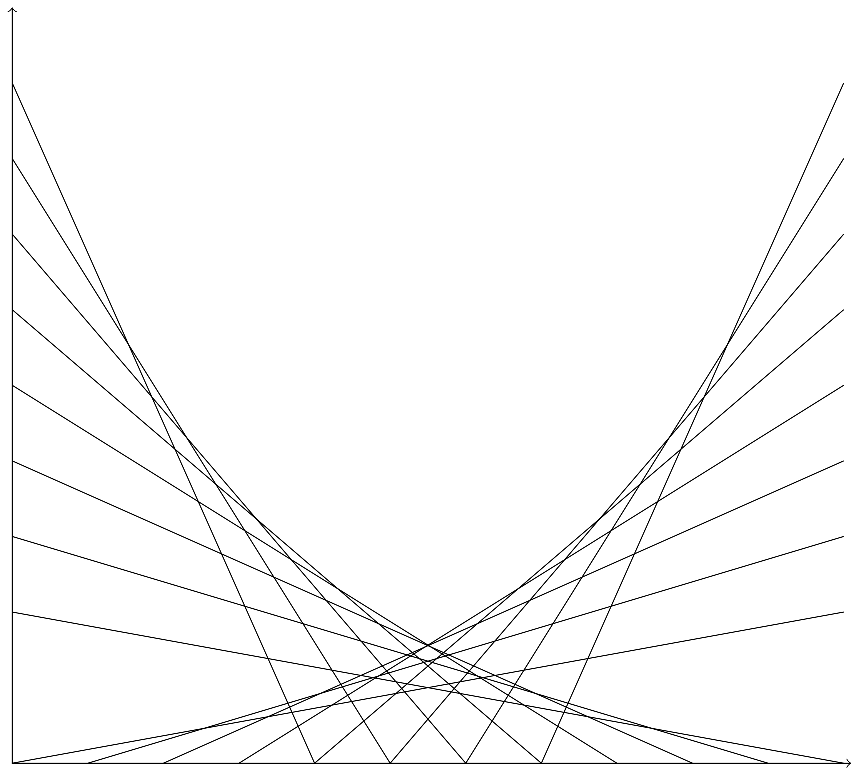
Optimization Handbook



Henri Lefebvre

June 7, 2019

Contents

Chapter 1

Introduction to Optimization

[4]

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 matrials 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 :

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

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

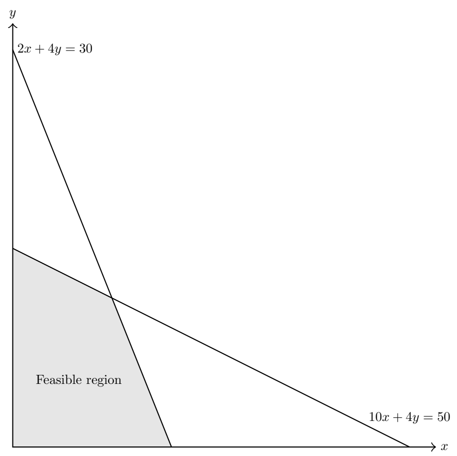


Figure 2.1: 2D representation of a feasible region

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 place, 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*

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

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 as 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 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 enouce the theorem, we need to introduce the concept of *basic* solutions.

Consider a feasible region defined as

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

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 $\{x \in \mathbb{R} | Ax = n, x \geq 0\}$ 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 :*

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

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) Let x be a feasible solution and let us denote by $a_1, ..., a_n$ the columns of A. Since it is feasible, the following holds :

$$a_1 x_1 + ... + a_n x_n = b$$

Assume now that exactely p variables x_i are greater than zero. For the sake of exposure, we'll assume that they correspond to the p first variables. That is :

$$a_1 x_1 + ... + a_p x_p = b$$

Then two different situations may occur :

$(a_1, ..., a_p)$ **are linearly independent** : then clearly we have $p \leq m$. If $p = m$, then x is a basic feasible solution and the proof is complete. If $p < m$ then since A is of full rank, one can find $m - p$ columns of A which forms a basis when added to the p columns $a_1, ..., a_p$. By setting $x_i = 0$ for all $i > p$, we obtain a (degenerate) basic feasible solution.

$(a_1, ..., a_p)$ **are linearly dependent** : Then there exists coefficients $\lambda_i, ..., \lambda_p$ all non-zero, at least one of which can be assumed to be positive, such that

$$a_1 \lambda_1 + ... + a_p \lambda_p = 0$$

Let $\varepsilon \in \mathbb{R}$, Since $Ax = b$ holds, it also holds that

$$(x_1 - \varepsilon\lambda_1)a_1 + ... + (x_p - \varepsilon\lambda_p)a_p = b \quad (2.1)$$

For $\varepsilon = 0$, this reduces to the original feasible solution. As ε increases, the different components may increase, decrease or remain constant depending on the sign of λ_i. Since we assumed that at least one λ_i is positive, then at least one component will decrease as ε increases. We increase ε to the first point where one or more components become zero. That is, we choose

$$\varepsilon = \min\{x_i/\lambda_i : \lambda_i > 0\}$$

For this specific value, the solution built in 2.1 is feasible and has at most $p-1$ positive variables. Repeating this process as much as necessary, we can eliminate positive variables untill we obtain a feasible solution with corresponding columns which are linearly dependent.

(ii) Let us consider an optimal solution x^* and, as in proof of (i), let us suppose that there are exactely p positive variables. Again, two cases may occure as to whether the selected columns of A are linearly dependent or not. If the columns are independent, the proof goes as for (i). If they are dependent, we still can apply the idea of the proof of (ii) yet we need to check that the objective value of $x^* - \varepsilon\lambda$ remains optimal. Note that the objective value of the constructed solution is

$$c^T x^* + \varepsilon c^T \lambda$$

By contradiction, suppose that $c^T\lambda \neq 0$. Then one can find a sufficiently small value for ε so that $c^T x^* + \varepsilon c^T \lambda < c^T x^*$ which contradicts the optimality of x^*. Hence, $c^T y = 0$ and the obtained solution is optimal.

□

Before stating an equivalence result previously mentioned between basic variables and extreme points of polyhedra, let us take a small detour with a geometrical example of the idea of the proof of theorem 1, point *(i)*. For that purpose, let us consider the following feasible region :

$$K = \left\{ x \in \mathbb{R}^3_+ \middle| \begin{array}{ll} x_1 + x_2 + x_3 & = 1 \\ x_2 + x_3 & = \frac{2}{3} \end{array} \right\}$$

K is depicted in figure 2.2. As well, a (non-basic) feasible point is depicted with coordinates $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Indeed, one can easily check that

$$\frac{1}{3}\begin{pmatrix}1\\1\end{pmatrix} + \frac{1}{3}\begin{pmatrix}1\\0\end{pmatrix} + \frac{1}{3}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1\\\frac{2}{3}\end{pmatrix}$$

Yet, clearly, this collection of vectors are linear dependent, in particular we have :

$$1\begin{pmatrix}1\\1\end{pmatrix} + 0\begin{pmatrix}1\\0\end{pmatrix} + (-1)\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$$

From the two above equations, we obtain by multiplying the second one by a scalar $\varepsilon > 0$ and subtracting the first one :

$$\left(\frac{1}{3} - \varepsilon\right)\begin{pmatrix}1\\1\end{pmatrix} + \frac{1}{3}\begin{pmatrix}1\\0\end{pmatrix} + \left(\frac{1}{3} + \varepsilon\right)\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1\\\frac{2}{3}\end{pmatrix}$$

If we choose $\varepsilon = \frac{1}{3}$ (i.e., $\min\{x_i/\lambda_i : \lambda_i > 0\}$), we find that $x = (0, \frac{2}{3}, \frac{1}{3})$ is also a feasible solution and it has one coefficient set to zero. Since we have two constraints ($m = 2$), this solution is a basic solution (also depicted in figure 2.2).

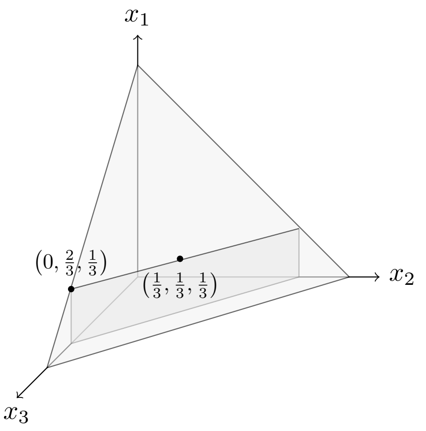

x_1
x_2
x_3
$(0, \frac{2}{3}, \frac{1}{3})$
$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

Figure 2.2: 3D representation of feasible region K

We can see that our basic feasible solution is indeed an extreme point of the polytope K. The following theorem states that this remark holds in general.

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.

$\Rightarrow$: Let x be a basic feasible solution for $\{x|Ax = b, x \leq 0\}$ where A is a $m \times n$ matrix. We have

$$a_1x_1 + ... + a_mx_m = b$$

By contradiction, let us suppose that there exists two different points $y, z \in \{x|Ax = b, x \geq 0\}$ such that x is a convex combination of these points, i.e., :

$$x = \alpha y + (1 - \alpha)z \quad 0 < \alpha < 1$$

Since $x \geq 0$ and α is positive, it holds that the last $n - m$ components of y and z are also equal to zero. Thus :

$$y_1a_1 + ... + y_ma_m = b$$
$$z_1a_1 + ... + z_ma_m = b$$

Yet since $a_1, ..., a_m$ are linearly independent, it follows that $x = y = z$ which contradicts that $y \neq z$. Hence, x is an extreme point.

$\Leftarrow$: Let x be an extreme point of $\{x|Ax = b, x \geq 0\}$ and let us assume that the nonzero components of x are the first k components, i.e.,

$$a_1x_1 + ... + a_kx_k = b$$

Let us show that $a_1, ..., a_k$ are linearly independent (i.e., that x is a basic feasible solution). By contradiction, suppose that $a_1, ..., a_k$ are linearly dependent. Then, there exists a vector $\lambda \geq 0$ with at least one non-zero coefficient such that

$$\lambda_1a_1 + ... + \lambda_ka_k = 0$$

Since $x \geq 0$, one can find a value for ε such that

$$x + \varepsilon\lambda \geq 0 \quad x - \varepsilon\lambda \geq 0$$

Yet, if this were the case, it would holds that $x = \frac{1}{2}(x + \varepsilon\lambda) + \frac{1}{2}(x - \varepsilon\lambda)$ which is a convex combination of two distinct feasible points. This contradicts the fact that x is an exterme point. Hence, x is a basic feasible solution.

□

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 nonzero. We therfore 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.

Observation 1. *The fundamental theorem of linear programming reduces the search space for optimality to the set of extreme points of the feasible region. Yet, note that for a problem with n variables and m constraints, we have at most*

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

extreme points (or equivalently, basic solutions).

3 Examples

3.1 Minimum cost flow problem

3.2 Support Vector Machine

3.3 Knapsack problem

Chapter 3

The Simplex algorithm

In chapter 2, we showed with the fundamental theorem of linear programming that the search space for optimality was reduced indeed to the set of extreme points of the feasible region, or equivalently of basic feasible solutions for the problem. The idea of the Simplex algorithm is to go from one extreme point to another in such a way that the objective function is always improved (i.e., decreases in case of a minimization). By assumption, the considered polyhedron is lower bounded (upper bounded in case of maximization) and therefore, the algorithm reaches the optimal point. The method can be thought of as a travel from an original extreme point to the one which maximizes the objective function.

The name Simplex comes from the name of the geometrical generalization of triangles to higher dimensions. Its name comes from the idea that it is the "simplest" closed geometrical object in n dimension.

The first section derives the algorithm formally. Then we explain in more details the geometrical interpretation of the Simplex algorithm. Since the algorithm starts with an initial basic feasible solution, the next section will explain how such a point can be found by using the same Simplex algorithm. In the three last sections, we give the pseudo code of the Revised Simplex algorithm written in matrix form and introduce two variants of the Simplex : (1) the bounded simplex which deals with bounded variables and (2) the transportation Simplex which is used for transportation problems.

1 Formal derivation

1.1 Assumptions

For the sake of demonstrations, we introduce the following assumption :

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

1.2 Pivoting and Gauss reduction

In this section, we explain how one can move from one basic solution to another with a standard operation from linear algebra called pivoting. This operation is used in Gauss method for solving a system of linear equations. Consider the following set of equations :

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ \vdots \qquad\qquad \ddots \qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m \end{array}$$

Of course, it is well known that if $m < n$ and if the equations are not redondant (i.e., they are linearly independent) there is not a unique solution. Yet, following the princple of pivoting from the Gauss elimination technique (see appendix A) one can turn this system in a so-called *canonical form* expressed as

$$\begin{array}{ccccccccc}
x_1 & + & \overline{a}_{1(m+1)}x_{m+1} & + & ... & + & \overline{a}_{1n}x_n & = & \overline{b}_1 \\
x_2 & + & \overline{a}_{2(m+1)}x_{m+1} & + & ... & + & \overline{a}_{2n}x_n & = & \overline{b}_2 \\
x_3 & + & \overline{a}_{3(m+1)}x_{m+1} & + & ... & + & \overline{a}_{3n}x_n & = & \overline{b}_3 \\
\vdots & & \vdots & & \ddots & & & & \vdots \\
x_m & + & \overline{a}_{m(m+1)}x_{m+1} & + & ... & + & \overline{a}_{mn}x_n & = & \overline{b}_m
\end{array}$$

which we often write, for the sake of synthesis, in a so-called *tableau* :

x_1	x_2	x_3	...	x_m	x_{m+1}	...	x_n	
1	0	0	0	0	$\overline{a}_{1(m+1)}$	...	$\overline{a}_{1n}$	$\overline{b}_1$
0	1	0	0	0	$\overline{a}_{2(m+1)}$	...	$\overline{a}_{2n}$	$\overline{b}_2$
0	0	1	0	0	$\overline{a}_{3(m+1)}$	...	$\overline{a}_{3n}$	$\overline{b}_3$
			$\ddots$		$\vdots$	$\ddots$		$\vdots$
0	0	0	0	1	$\overline{a}_{m(m+1)}$	...	$\overline{a}_{mn}$	$\overline{b}_m$

It is clear that if one posesses a system of equations written in cannonical form, then the solution given by $x = (\overline{b}_1, \overline{b}_2, ..., \overline{b}_m, 0, 0, ..., 0)$ is a basic solution. The idea of the Simplex algorithm is, in fact, to pivot from one canonical form to another. The question however is how to select the variable which will enter the basis and which one will leave the basis in order to increase a given objective function. This is deatailed in the following sub-sections.

1.3 Vector leaving the basis

First approach

In this section, we show how one can decide which variable should leave the basis. In fact, in the previous section, we showed that the pivot operation allows us to move from a basic solution to another, however, such a move does not guarantee the feasibility of the obtained basic solution. In other words, it is not established that the pivot operation will keep the positivity of the variables. We present here a sufficient condition for the pivot operation to keep the feasibility property when moving from one basic solution to another.

Let x be a basic solution. We have

$$a_1x_1 + a_2x_2 + ... + a_mx_m = b$$

where $x_i > 0, \forall i = 1...m$ (nondegeneracy assumption). And suppose that we want to bring x_q in the basis. The question is how to choose which variable has to leave the basis in order to keep feasibility of the new basic solution. For that purpose, let us write a_q in terms of the current basis :

$$a_q = \lambda_1 a_1 + \lambda_2 a_2 + ... + \lambda_m a_m$$

From the two above equalities, we derive the following :

$$(x_1 - \varepsilon\lambda_1)a_1 + (x_2 - \varepsilon\lambda_2)a_2 + ... + (x_m - \varepsilon\lambda_m)x_m + \varepsilon a_q = b$$

for any $\varepsilon > 0$, which is a linear combination of at most $m+1$ vectors. Setting $\varepsilon = 0$ yields the current basic feasible solution. As ε increases, the coefficient of a_q increases. Yet, it yields a non basic variable, in general. The coefficients of the other vectors may increase or decrease with ε depending on the original coefficients (i.e., if we have $\lambda_i > 0$). Therefore, by taking the first value of ε which makes vanishing such a vector, we ensure the feasibility of the obtained new basic solution. Formally, we choose ε such that :

$$\varepsilon = \min\{x_i/\lambda_i : \lambda_i > 0\}$$

If the minimum is achieved by more than one variable, the new basic feasible solution is degenerate. If, however, none of the λ_i are positive, this means that all the coefficients increase as ε increases, without restriction, while keeping feasibility. This corresponds to a case where the polyhedron is unbounded.

Alternative approach

Another approach for reaching the same result is to consider a problem in standard form and to notice that we can always write a basic feasbile solution like so :

$$\overline{b}_i - \sum_{j=m}^{n} \overline{a}_{ij}x_j \geq 0 \qquad \forall i = 1, ..., m$$

We want to increase a non-basic variable x_q while keeping feasibility. Also, we'd like to increase x_q as much as we can. That is, we keep the other non-basic solutions to zero. This yields :

$$\overline{b}_i \geq \overline{a}_{ij}x_q \qquad \forall i = 1, ..., m$$

Thus, it follows that :

$$\begin{cases} \dfrac{\bar{b}_i}{\bar{a}_{ij}} \geq x_q & \forall i = 1, ..., m | \bar{a}_{ij} > 0 \\ \dfrac{\bar{b}_i}{\bar{a}_{ij}} \leq x_q & \forall i = 1, ..., m | \bar{a}_{ij} < 0 \end{cases}$$

Note that when $\bar{a}_{ij} < 0$, no upper bound on x_q can be derived. Hence, when all the a_{ij} coefficients are negative, the problem is unbounded. On the contrary, when at least one coeeficient is negative, we do have a constraint on x_q and to ensure feasibility with respect to all the constraints, one should increase x_q by

$$\min \left\{ \frac{\bar{b}_j}{\bar{a}_{ij}} : \bar{a}_{ij} > 0 \right\}$$

which implies setting x_k to zero where k denotes the argmin of the above finite minimum. We get is the same result as in the first approach.

1.4 Vector entering the basis

In the previous section, we have shown how to choose which variable had to leave in order to keep feasibility when we want to insert a given variable in the basis. In that sense, it allows us to travel from one basic solution to another while keeping feasibility. In this section, we show how to choose which variable should enter the basis in order to increase a given objective function. For that purpose, consider the following objective function :

$$c^T x = c_1 x_1 + c_2 x_2 + ... + c_n x_n$$

And consider a feasible basic solution $\hat{x} = [\hat{x}_B, 0]$. Its objective value is given by

$$c_1 \hat{x}_1 + c_2 \hat{x}_2 + ... + c_m \hat{x}_m$$

If we write our system of equalities in canonical form, recall that we trivially have $x_j = \bar{b}_j$ for basic variables and $x_j = 0$ for non-basic variables. The key idea here, is to express the objective function value of a general solution x in terms of the objective value of the basic solution $\hat{x}$. First note that the following holds for any feasible solution x :

$$\begin{aligned} x_1 &= \bar{b}_1 - \sum_{j=m+1}^{p} \bar{a}_{1j} x_j \\ x_2 &= \bar{b}_2 - \sum_{j=m+1}^{p} \bar{a}_{2j} x_j \\ &\vdots \qquad \vdots \\ x_m &= \bar{b}_m - \sum_{j=m+1}^{p} \bar{a}_{mj} x_j \end{aligned}$$

By substituting the basic variables from $\hat{x}$ by the above equalities, we get :

$$\begin{aligned} c^T x &= \sum_{j=1}^{n} c_j x_j \\ &= \sum_{j=1}^{m} c_j x_j + \sum_{j=m+1}^{n} c_j x_j \\ &= \sum_{j=1}^{m} c_j \left(\bar{b}_j - \sum_{k=m+1}^{n} \bar{a}_{jk} x_k \right) + \sum_{j=m+1}^{n} c_j x_j \\ &= \sum_{j=1}^{m} c_j \bar{b}_j - \sum_{j=1}^{m} \sum_{k=m+1}^{n} c_j \bar{a}_{jk} x_k + \sum_{j=m+1}^{n} c_j x_j \\ &= \sum_{j=1}^{m} c_j \bar{b}_j - \sum_{k=m+1}^{n} x_k \left(\sum_{j=1}^{m} c_j \bar{a}_{jk} \right) + \sum_{j=m+1}^{n} c_j x_j \\ &= \sum_{j=1}^{m} c_j \hat{x}_j + \sum_{j=m+1}^{n} x_j \left(c_j - \sum_{i=1}^{m} \bar{a}_{ij} c_i \right) \\ &= c_B^T \hat{x}_B + \sum_{j=m+1}^{n} (c_j - z_j) x_j \end{aligned}$$

where

$$z_j = \sum_{i=1}^{m} \bar{a}_{ij} c_i$$

This result gives us a condition for a vector to benificially enter the basis. Indeed, if, for a given variable j, we have $c_j - z_j < 0$ then it means that increasing the value of x_j from zero to a positive value will decrease the objective function. Hence, going from the solution $\hat{x}$ to another solution which includes $x_j > 0$ will yield a lower value of the objective.

We introduce the standard notation $r_j = c_j - z_j$. These coefficients are called *reduced cost* or *relative costs* since they measure the cost of a variable respectively to a given basis. We can interpret these numbers as the gain we would obtain to use a real variable x_j instead of the linear combination giving $x_j = \sum_{j=1}^{m} \overline{a}_j x_j$. Another interpretation is to see the reduced cost as the amount by which the objective cost would have to decrease (for minimization problems) in order to make the entrance of a column profitable.

We now can state the two following theorems :

Theorem 3. *Given a non-degenerate basic feasible solution with corresponding objective value z_0. If there exist a column such that $c_j - z_j < 0$, then there is a feasible solution with objective value $z < z_0$. If the column $\overline{a}_j$ can be substituted for some vector in the original basis to yield a feasible basic solution, then this solution will have an objective value $z < z_0$. If, however, $\overline{a}_j$ cannot be substituted to yield a basic feasible solution, the problem is unbounded and the optimal solution tends to minus infinity.*

Theorem 4 (Optimality condition)**.** *If, for some basic feasible solution, $c_j - z_j \geq 0$ for all j, then the solution is optimal.*

2 Finding a feasible solution

As previously explained the Simplex procedure needs an initial feasible point to start. A practical way to obtain one is by solving an optimization problem whose solution is an extreme point of the feasible region and for which we do know a feasible solution. This step is called the Phase I of the Simplex. The idea is to introduce *artificial* variables which are not part of the original problem and to replace the original objective by the minimization of a positive sum of the artifical variables. Clearly, since we do not constrain the artifical variables, a feasible solution for that problem is to select all the artifical variables and to set the rest to zero. Its objective value is simply the number of added artifical variables. Solving this problem, if a feasible solution exists, the Simplex will find one since it would not contain any artifical variable and its cost would therefore be minimal.

Formally, to find a feasible solution for the feasible region $\{x \in \mathbb{R}_+^n | Ax = b\}$, one has to solve the folowing problem

$$\begin{aligned} \text{minimize } & \sum_{k=1}^{m} x_{n+k} \\ \text{s.t. } & Ax = b \\ & x_j \geq 0 \quad \forall j = 1...n+m \end{aligned}$$

For which an initial feasible solution is clearly given by $(\underbrace{0, \;\; ..., \;\; 0}_{n \text{ original variables}} \;, \; \underbrace{1, \;\; ..., \;\; 1}_{m \text{ artifical variables}})$.

3 The Simplex method

3.1 Pseudo-code

The Simplex method takes as input a feasible basic solution. We assume that we start the algorithm with a canonical form of $Ax = b$. We append a row at the bottom of this tableau, obtained the so-called *Simplex Tableau*, in which we report the reduced costs and the objective function in the corner. The Simplex Tableau is presented in table 3.1.

x_1	x_2	...	x_m	x_{m+1}	...	x_n	b
1	0	...	0	$\overline{a}_{1(m+1)}$	...	$\overline{a}_{1n}$	$\bar{b}_1$
0	1	...	0	$\overline{a}_{2(m+1)}$	...	$\overline{a}_{2n}$	$\bar{b}_2$
$\vdots$		$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
0	0	...	1	$\overline{a}_{m(m+1)}$	...	$\overline{a}_{mn}$	$\bar{b}_m$
0	0	...	0	r_{m+1}	...	r_n	$-z_0$

Table 3.1: The Simplex Tableau

Regarding the last row of the tableau, we may interpret it as z being an extra variable subject to the constraint $z = \sum_{j=1}^{n} c_j x_j$. A basic solution would therefore be of size $m+1$, yet we can require that z must be part of it. For that reason, it is not necessary to add a column in the tableau for z since it would always be equal to $(0, 0, ..., 0, 1)$. Therefore, initially, the last row contains the objective cost of each variables and a right hand side of zero. By pivoting in order to put the system into standard form, the last row's values for basic variables can also be reduced to zero. This is, in fact, equivalent to transforming $c^T x = z$ into

$$r_{m+1}x_{m+1} + r_{m+2}x_{m+2} + ... + r_n x_n - z = -z_0$$

Where r_j denotes the reduced cost of variable x_j. The Simplex algorithm is stated in 1.

Algorithm 1 Simplex Algorithm

Step 0 : From a system in canonical form, compute the reduced costs by row reduction on the last row

Step 1 : If each $r_j \geq 0$, STOP. The solution is optimal.

Step 2 : Find q such that $r_q < 0$ to determine which variable will enter the basis.

Step 3 : Find the variable with corresponding minimum ratio $\min\{\bar{b}_i/\bar{a}_{iq} : \bar{a}_{iq} > 0\}$. Let p be the index of that variable.
If there is no coefficient $\bar{a}_{iq} > 0$, STOP. The problem is unbounded, the solution in $-\infty$.

Step 4 : Pivot every rows (including the last one) on the pqth element. Go to **Step 1**.

4 Degeneracy

Running the Simplex algorithm, it is possible to find degenerate basic feasible solutions. In most cases, these degenerate solutions can be handled in the same way as any basic feasible variable. Yet, in some situations, it is that, having a current degenerate feasible basic solution, one computes the variable to enter the basis and the associated minimum ratio to be zero. In this case, a zero valued basic variable will enter the basis while not improving the objective function. In this case, it may occur that the Simplex enters in an infinite cycle, going from one degenerate solution to another degenerate solution.

This section is devoted to the study of such cases. First, we give an example of Simplex cycling, then, give some pivot rules to avoid cycling.

4.1 Example

The example we give here was introduced in [3]. TODO

4.2 Pivot Rules

Dantzig's Rule

Bland's Rule

Steepest-edge rule

Approximate steepest-edge rule

Partial pricing

TODO

4.3 Examples

Optimal solution

Le us consider the following problem :

$$\begin{aligned} \text{maximize } & x_1 + x_2 \\ \text{s.t. } & 4x_1 + x_2 \leq 20 \\ & x_1 + 2x_2 \leq 11 \\ & x_1, x_2 \geq 0 \end{aligned}$$

The first step to do is to write this problem in standard form by introducing two slack variables s_1 and

s_2 and to turn this maximization problem into a minimization problem like so :

$$\begin{aligned} \text{minimize } & -x_1 - x_2 \\ \text{s.t. } & 4x_1 + x_2 + s_1 = 20 \\ & x_1 + 2x_2 + s_2 = 11 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned}$$

This gives us a good Simplex Tableau to start with. Indeed, the slack variables already form a basis to start with and setting $s_1 = 20$ and $s_2 = 11$ is a basic feasible solution. The initial tableau is :

x_1	x_2	s_1	s_2	b
4	1	1	0	20
1	2	0	1	11
-1	-1	0	0	0

Where we see that variables x_1 and x_2 are elligeable to enter the basis (i.e., have negative reduced costs). We'll use the "most negative reduced cost" rule for pivoting, i.e., the variable which leaves the basis corresponds to the one having the most negative reduced cost. In our case, both negative-reduced-cost variables have a reduced cost of -1. We arbitraly use x_2. Then, we compute the ratios for positive values of the column $\bar{b}_i/\bar{a}_{i2}$. The minimum one is $\min(20/1; 11/2) = 5.5$ and is reached in the second row. Thus, s_2 exits the basis. After pivoting on $2, 2$, the resulting tableau is :

x_1	x_2	s_1	s_2	b
$\frac{7}{2}$	0	1	$-\frac{1}{2}$	$\frac{29}{2}$
$\frac{1}{2}$	1	0	$\frac{1}{2}$	$\frac{11}{2}$
$-\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{11}{2}$

Here, only variable x_1 has a negative reduced cost. Thus, x_1 will enter the basis. Naturally, s_1 will leave the basis since $\min(\frac{29}{2}/\frac{7}{2}; \frac{11}{2}/\frac{1}{2}) = \frac{29}{7}$. This leads to the following table :

x_1	x_2	s_1	s_2	b
1	0	$\frac{2}{7}$	$-\frac{1}{7}$	$\frac{29}{7}$
0	1	$-\frac{1}{7}$	$\frac{4}{7}$	$\frac{24}{7}$
0	0	$\frac{1}{7}$	$\frac{3}{7}$	$\frac{53}{7}$

Here, every non basic reduced cost are positive, the Simplex algorithm terminates and the solution is optimal. Figure 3.1 depicts the travel made by the Simplex algorithm throughout its execution. The initial basic solution is $(0, 0)$, then goes to $(0, 11/2)$ and finally reaches the optimal solution $(29/7, 24/7)$.

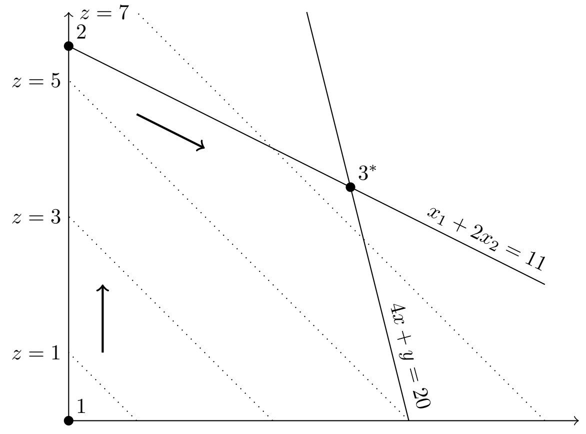


Figure 3.1: Example of Simplex moves

Unbounded problem

In this section, we consider the optimization problem stated below. It is our intention that the problem is unbounded, i.e., its optimal solution goes to infinity. The goal of this example is to show how the Simplex algorithm terminates in such situations and what are the informations which are still usable in such cases. Consider the following :

$$\begin{aligned} \text{maximize } & 2x_1 + x_2 \\ \text{s.t. } & 2x_1 - x_2 \geq -5 \\ & x_1 - x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Clearly, the feasible region is unbounded as it can be seen in figure 3.2. Note that the Simplex works even with unbounded polyhedron and is able to compute optimal solution in such cases. The specificty of this problem is that the plane defining the objective function is "oriented" (i.e., is increasing) towards the unbounded direction. In chapter 10, we give a sufficient condition for a problem to be bounded implying the extreme rays of a polyhedron. In this case, the objective function and the extreme rays of the feasible region are oriented in the same direction.

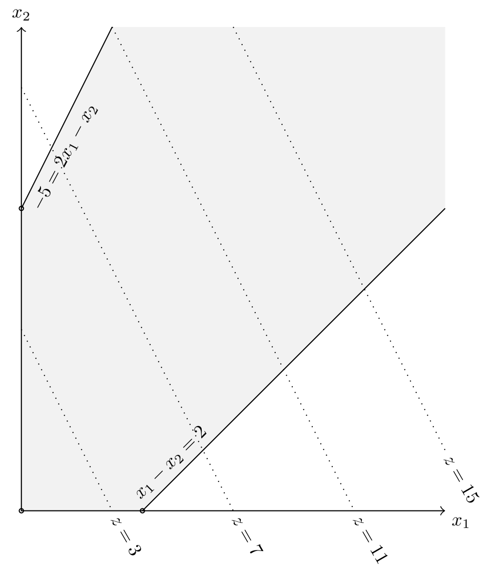


Figure 3.2: Unbounded feasible region

First, let us write the considered problem in standard form by introducing the slack variables and turning the maximization problem into a minimization problem, this gives :

$$\begin{aligned} \text{minimize } & -2x_1 - x_2 \\ \text{s.t. } & -2x_1 + x_2 + s_1 = 5 \\ & x_1 - x_2 + s_2 = 2 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned}$$

We introduce the Simplex tableau of this problem :

x_1	x_2	s_1	s_2	b
-2	1	1	0	5
1	-1	0	1	2
-2	-1	0	0	0

Looking at the most negative reduced cost, we determine that x_1 will enter the basis and by, computing the minimum ratios, that s_2 will leave the basis. We get :

x_1	x_2	s_1	s_2	b
0	-1	1	2	9
1	-1	0	1	2
0	-3	0	2	2

Again, looking at the most negative reduced cost, we can see that x_2 should enter the basis. However, there are no positive coefficient in that row. This means that we can increase x_2 as much as we want while keeping feasibility (i.e., there are no feasibility restrictions). Thus, we conclude that the problem is unbounded. Yet, how can we interpret the vector $(-1,-1)^T$ returned by the Simplex ? Answer : it is the opposite direction an the extreme ray of the feasible region.

$$d = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Reduced costs

Computation In this section, we suppose having a canonical tableau of the problem from the first example :

x_1	x_2	s_1	s_2	b
$\frac{7}{2}$	0	1	$-\frac{1}{2}$	$\frac{29}{2}$
$\frac{1}{2}$	1	0	$\frac{1}{2}$	$\frac{11}{2}$

where the reduced costs are missing. We show by example how to compute the reduced costs.

First, note that (s_1, x_2) is a basis of $\mathbb{R}^2$ $(m = 2)$. Thus, the associated reduced costs are 0. Concerning the non-basic variables, we can compute the reduced costs using the formula $r_j = c_j - z_j$ where $z_j = \sum_{i=1}^m \overline{a}_{ij} c_i$. We recall here the objective function : $c^T x = -x_1 - x_2$. Therefore, we have :

$$\begin{aligned} r_{x_1} &= -1 - \sum_{i=1}^m \overline{a}_{i1} c_i = -1 - \left(\frac{7}{2} \times 0 + \frac{1}{2} \times -1 \right) \\ &= -\frac{1}{2} \end{aligned}$$

$$r_{s_2} = 0 - \sum_{i=1}^m \overline{a}_{i1} c_i = 0 - \left(-\frac{1}{2} \times 0 + \frac{1}{2} \times -1 \right)$$

$$= \frac{1}{2}$$

Finally, we easily compute the objective value :

$$z = c^T x = -1 \times 0 + (-1) \times \frac{11}{2} + 0 \times \frac{29}{2} + 0 \times 0 = -\frac{11}{2}$$

Interpreation In this short example, we show the interpretation of the reduced costs as the increase in the variable's objective cost necessary to make the entrance of the variable in the basis profitable. Indeed, consider the (very dumb) following problem :

$$\begin{aligned} \text{maximize } & 4x_1 + x_2 \\ \text{s.t. } & x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

First, let us write this problem in standard form by turning the maximization into a minimization problem and by introducing slack variables :

$$\begin{aligned} \text{minimize } & -4x_1 - x_2 \\ \text{s.t. } & x_1 + x_2 + s = 1 \\ & x_1, x_2, s \geq 0 \end{aligned}$$

The feasible region of the problem is depicted in figure 3.3 in the top plot as well as the levels of the objective function. We can easily see that the optimal solution (marked by a $*$ in the plot) is $(1, 0)$. Let us run the Simplex algorithm for that problem :

$$\begin{array}{ccc|c} x_1 & x_2 & s & \\ \hline 1 & 1 & 1 & 1 \\ \hline -4 & -1 & 0 & 0 \end{array} \rightarrow \begin{array}{ccc|c} x_1 & x_2 & s & \\ \hline 1 & 1 & 1 & 1 \\ \hline 0 & 3 & 4 & 4 \end{array}$$

First, we look for the most negative reduced cost. This indicates that x_1 will enter the basis. Since we are in one dimension, the basis is composed of a single element. Therefore, the leaving variable is simply the variable forming the current basis, i.e., s. In two steps, the Simplex terminates and we can see that the reduced cost of variable x_2 is $r_2 = 3$. This means that, if we increase the cost c_2 of at least x_2 in the objective function, it may be profitable to include x_2 in the basis. Indeed, in the second plot from figure 3.3, we depicted the objective function with an increase in c_2 of exactely r_2. We can see that the objective function reaches its maximum value simultaneously in $(0, 1)$ and in $(1, 0)$. In the last plot however, we increased even more the cost c_2 for variable x_2 while keeping the other costs constant. We can see that now, the optimal solution is realised in $(0, 1)$.

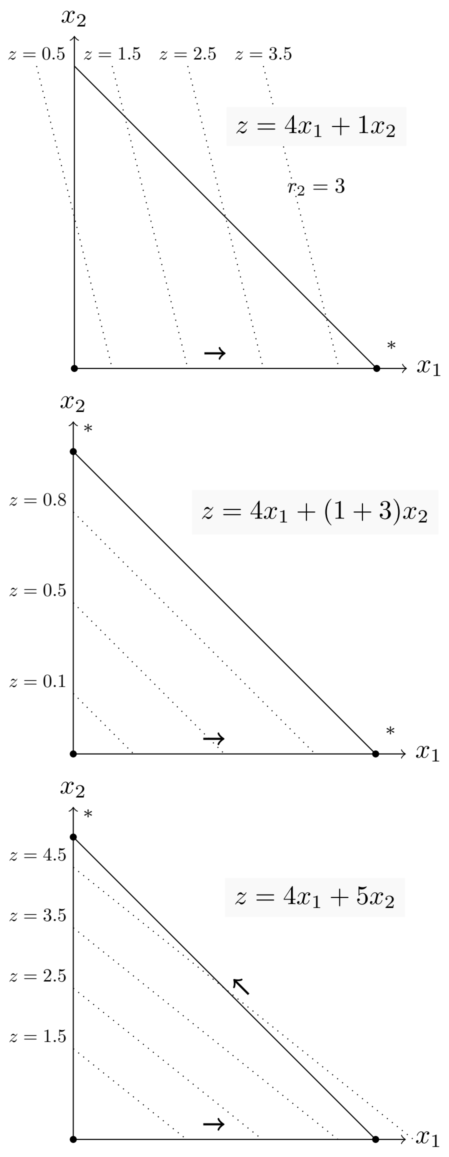


Figure 3.3: Interpretation of the reduced cost

Chapter 4

The Simplex algorithm (2)

In this chapter, we present two variants of the Simplex algorithm. The first variant is the so-called *Revised Simplex Algorithm* which is a more efficient version of the Simplex in practise from a computational point of view. The *Bounded Simplex* is then introduced as a way to solve problems in which some variables are upper bounded without introducing an explicit constraint. Examples for both variants are discussed.

1 The revised Simplex

1.1 Matrix form of the Simplex

We present here, the matrix formulation of the original Simplex method. First, note that the tableau from the original Simplex is solely characterized by the variables included in the current basis. In that sense, knowing the basic variables implies being able to compute the associated tableau. We consider the matrix constraint A and denote by B the matrix formed by the columns of B corresponding to the basic variables. Again, we assume that they correspond to the first m columns for the sake of notation. C will denote the remaining $n-m$ columns aggregated in a matrix. Thus, we have :

$$A = [A, B]$$

$$x = [x_B, x_C] \qquad c^T = [c_B^T, c_D^T]$$

Using these notations, a problem written in standard form becomes :

$$\begin{aligned} \text{minimize } & c_B^T x_B + c_D^T x_D \\ \text{s.t. } & Bx_B + Dx_D = b \\ & x_B, x_D \geq 0 \end{aligned}$$

The basic solution x which corresponds to the basis B is given by $x = (x_B, 0)$ where we have $x_B = B^{-1}b$, obtained by setting $x_D = 0$. In general, however, the following result holds :

$$x_B = B^{-1}b - B^{-1}Dx_D$$

which gives the expression of the objective function to be

$$\begin{aligned} z &= c_B^T(B^{-1}b - B^{-1}Dx_D) + c_D^T x_D \\ &= c_B^T B^{-1}b + (c_D^T - c_B^T B^{-1}D)x_D \end{aligned}$$

which gives the matricial expression for the reduced costs :

$$r_D^T = c_D^T - c_B^T B^{-1}D$$

Therefore, the initial tableau is given by :

$$\begin{array}{cc|c} B & D & b \\ \hline c_B^T & c_D^T & 0 \end{array} \longrightarrow \begin{array}{cc|c} I & B^{-1}D & B^{-1}b \\ \hline 0 & c_D^T - c_B^T B^{-1}D & -c_B^T B^{-1}b \end{array}$$

1.2 Pseudo-code

Even though the Simplex is of exponential complexity, experience has how an average polynomial complexity in $\mathcal{O}(m)$. To that extent, pivoting can be regarded as an operation which is not done "that often". However, if we consider cases where $n >> m$, pivoting implies computing numerous coefficient which in turn may be useless. The Revised Simplex method avoids these useless computations. It is given in 2. It starts with a given feasible basis B.

Algorithm 2 Revised Simplex Algorithm

```
Step 1 : Compute the reduced costs $r_D^T = c_D^T - c_B^T B^{-1} D$ by computing $y^T = c_B^T B^{-1}$ then $r_D^T = c_D^T - y^T D$. If $r_D \geq 0$, STOP. The current solution $B^{-1}b$ is optimal.
Step 2 : Determine which variable $x_q$ enters the basis by using the pivot rule of your choice (e.g., most negative reduced cost). Compute $\overline{a}_q = B^{-1}a_q$ which gives column $a_q$ in terms of the current basis.
Step 3 : Compute $p = \text{argmin}\{\overline{b}/\overline{a}_{iq} : \overline{a}_{iq} > 0\}$. $p$ will leaves the basis. If no $\overline{a}_{iq} > 0$, STOP. The problem is unbounded.
Step 4 : Update $B^{-1}$. Return to Step 1.
```

Updating B^{-1} is done through the usual pivot operations on $[B^{-1}, \overline{a}_q]$.

Interstingly enough, it is not necessary to know B^{-1} thoughout the Simplex itations. In fact, we do need to solve linear systems with B as coefficient matrix. An LU-decomposition of B can be used to fasten the resolution (see appendix A for details on LU-decomposition).

1.3 Examples

TODO

2 The bounded Simplex

In this section, we present a variant of the Simplex algorihtm used when the problem contains bounded variables. Formally, we take interest in problems of the form :

$$\begin{aligned} \text{minimize } & c^T x \\ \text{s.t. } & Ax = b \\ & l_i \leq x_i \leq u_i \text{ for some } i \\ & x \geq 0 \end{aligned}$$

The idea of the bounded Simplex is to tweak the Simplex algorithm in order to *disregard* the bound constraints from the general computations. These constraints will be directly embedded within the algorithm.

2.1 Formal derivation

2.2 Pseudo-code

2.3 Pre-processing

2.4 Examples

3 The Network Simplex

3.1 Formal derivation

3.2 Pseudo-code

3.3 Examples

Chapter 5

Lagrangian duality

1 Motivations

2 Practical derivations

3 Duality theorems

4 Geometric interpretation

5 Sensitivity

6 Complentary slackness

7 The Primal-Dual algorithm

7.1 Examples

Chapter 6

The branch-and-bound algorithm

Chapter 7

The branch-and-cut algorithm

Chapter 8

Column generation and the branch-and-price algorithm

Chapter 9

Relaxation techniques

1 Formal definition

2 Linear relaxation

3 Lagrangian relaxation

4 Surrogate relaxation

Chapter 10

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 seperates 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 :

$$\text{minimize } c^T x + f(y) \tag{10.1}$$

$$\text{s.t. } Ax + g(y) = b \tag{10.2}$$

$$y \in Y, x \geq 0 \tag{10.3}$$

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 :

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

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

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

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

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

where $\{v^j\}_{j=1,...,J}$ 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 9). 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 $At \leq 0$. 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 λ, $z + \lambda v^k$ is a feasible solution for problem $(\mathcal{P})$. However, its associated objective value is given by $c^T u + \lambda c^T v^k \to +\infty$ when $\lambda \to +\infty$ 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 $c^T v^j \leq 0, \forall j = 1, ..., J$. Problem $(\mathcal{P})$ is therefore upper bounded by $\sum_i \alpha_i \max_i\{c^T u^i\}$. □

Theorem 5 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 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 10.1.

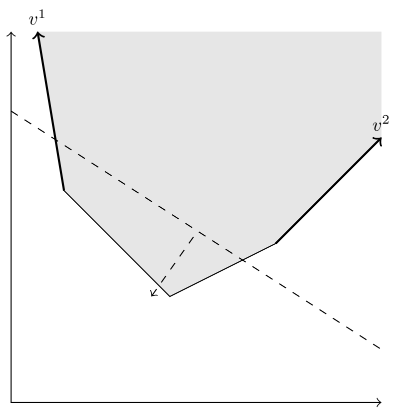


Figure 10.1: Illustration of theorem 5

If we apply theorem 5 to our specific problem, we find that a necessary and sufficient condition for $q(y)$ to be upper bounded is that $(b - g(y))^T v^j \leq 0, \forall j = 1, ..., J$ 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 $u^i, i = 1, ..., I$ the extreme points of the feasible region, the problem of finding the value of $q(y)$ can be reformulated as

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

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

$$\begin{aligned} \text{minimize } & f(y) + q && (10.4)\\ \text{s.t. } & y \in Y && (10.5)\\ & (b - g(y))^T u^i \leq q \quad \forall i = 1, ..., I && (10.6)\\ & (b - g(y))^T v^j \leq 0 \quad \forall j = 1, ..., J && (10.7) \end{aligned}$$

where constraints 10.6 are called *optimality cuts/constraints* since they define the extreme points of the feasible region and constraints 10.7 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 3.

3.2 Generating a feasibility cut

In **Step 2** of algorithm 3, 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 unbouned. 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 postive terms (i.e., no constraint bounds its value). The associated column is in fact an extreme ray of the polyhedron.

Algorithm 3 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 :
        minimize $f(y) + q$
            s.t. $y \in Y$
                $(b - g(y))^T u^i \leq q \quad \forall i = 1, ..., p$
                $(b - g(y))^T v^j \leq 0 \quad \forall j = 1, ..., k$
    Let $\bar{y}, \bar{q}$ be the optimal solution thus obtained.
Step 2 : Check the feasible 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 5, 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.
```

4 Examples

// Graphical representation

5 Stabilisation methods

5.1 Bundle methods

5.2 Proximal methods

6 Generalization

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

$$\begin{aligned} \text{minimize } & f(x,y) && (10.8)\\ \text{s.t. } & g(x,y) \le 0 && (10.9)\\ & x \in X && (10.10)\\ & y \in Y && (10.11)\end{aligned}$$

under the following hypothesis :

(i) X is a convex set

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

It was first introduced in [2]. The main aspects of the demonstrations are given in this section.

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

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

where we use the convention that $\inf\{h(x) : x \in A\} = +\infty$ if $A = \emptyset$.

Let us denote by $v(y)$ the minimization problem over x, i.e.,

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

And note that $v(y)$ is considered easy to solve (by definition of y as being complicating variables). Also, we will consider the following set corresponding to the feasible values of y allowing one or more feasible values for x :

$$Y \cap V = Y \cap \{y | \exists x \in X, g(x,y) \le 0\}$$

which turns our problem into :

$$\min\{v(y) : y \in Y \cap V\}$$

without needing any sort of convention on the infimum operator whatsoever.

The development of the generalized Benders decomposition now depends on our ability to describe $v(y)$ and V, and on our ability to check the feasibility/optimality of a solution. The first two points are solved with the two following theorems :

Theorem 6 (V-representation)**.** *Let X be a non-empty convex set and G be an n-dimensional convex-in-x function for all $y \in Y$. If $\{z \in \mathbb{R}^n | g(x,y) \le z \forall x \in X\}$ is closed for all $y \in Y$, then, for any $y \in Y$,*

$$y \in V \Leftrightarrow \inf\{\lambda^T g(x,y) : x \in X\} \le 0, \forall \lambda \in \Lambda$$

with $\Lambda = \{\lambda \in \mathbb{R}^n | \sum_{i=1}^n \lambda_i = 1, \lambda \ge 0\}$

Proof.

$\Rightarrow$: if $y \in V$ then, by definition, there exists an $x \in X$ such that $g(x,y) \le 0$ and therefore $\inf\{\lambda^T g(x,y) : x \in X\} \le 0, \forall \lambda \in \Lambda$.

$\Leftarrow$: if $\inf\{\lambda^T g(x,y) : x \in X\} \le 0, \forall \lambda \in \Lambda$ then

$$\sup\{\inf\{\lambda^T g(x,y)\} : \lambda \in \Lambda\} \le 0$$

which implies, since the scaling of $\lambda \in \Lambda$ does not influence the sign of the quantity of which we take the supremum, that

$$\sup\{\inf\{\lambda^T g(x,y)\} : \lambda > 0\} \le 0$$

And therefore,

$$\sup\{\inf\{\lambda^T g(x,y)\} : \lambda \ge 0\} = 0$$

Which is equivalent to saying that the dual of the following primal problem has an optimal value of zero :

$$\begin{aligned} \text{minimize } & 0^T x\\ \text{s.t. } & g(x,y) \le 0\end{aligned}$$

$$x \in X$$

The set $\{z \in \mathbb{R}^n | g(x,y) \le z \forall x \in X\}$ being a closed set is a sufficient condition for the primal problem to have a feasible solution. Hence, the fact that $y \in V$

□

Observation 2. *We give here some sufficient conditions for $\{z \in \mathbb{R}^n | g(x,y) \le z \forall x \in X\}$ to be closed for any $y \in Y$:*

- *X is closed and bounded and $g_i(x,y)$ are continuous over X for all $y \in Y$*
- *TODO*

The problem can therefore be written as

$$\begin{aligned} \text{minimize } & h(y) \\ \text{s.t. } & \inf_{x \in X} \{\lambda^T g(x,y)\} \le 0 \quad \forall \lambda \in \Lambda \\ & y \in Y \end{aligned}$$

The following theorem then reformulates problem $v(y)$:

Theorem 7 ($v(y)$-representation). *Let X be a nonempty convex set and assume that $f(x,y)$ and $g_i(x,y)$ are convex-in-x over X for any fixed $y \in Y$. Suppose also that for any $y \in Y \cap V$, the following holds :*

- *$h(y)$ is finite*
- *there exist optimal multipliers for the problem defining $v(y)$*

Then, for any $y \in Y \cap V$, the primal problem defining $v(y)$ has the same optimal value as its dual :

$$\sup_{u \ge 0} \{ \inf_{x \in X} f(x,y) + u^T g(x,y)\}$$

Proof. See Lagrangian duality. □

The problem can now be stated, by introducing a new variable y_0, as

$$\begin{aligned} \text{minimize } & y_0 \\ \text{s.t. } & \sup_{u \ge 0} \{ \inf_{x \in X} f(x,y) + u^T g(x,y)\} = y_0 \\ & \inf_{x \in X} \{\lambda^T g(x,y)\} \le 0 \quad \forall \lambda \in \Lambda \\ & y \in Y \end{aligned}$$

And, by definition of the sup operator, as

$$\begin{aligned} \text{minimize } & y_0 \\ \text{s.t. } & \inf_{x \in X} f(x,y) + u^T g(x,y) \le y_0 \quad \forall u \ge 0 \\ & \inf_{x \in X} \{\lambda^T g(x,y)\} \le 0 \quad \forall \lambda \in \Lambda \\ & y \in Y \end{aligned}$$

A simple way to solve this problem is through relaxation. We need, however, to be able to recognise situations in which the optimal solution is found or, in the contrary, to generate cuts.

Now, note that the following so-called *sub-problem* is easy to solve for a fixed value of $y = \bar{y}$:

$$\begin{aligned} h(\bar{y}) = \text{minimize } & f(x, \bar{y}) \\ \text{s.t. } & g(x, \bar{y}) \le 0 \\ & x \in X \end{aligned}$$

For which three different situations can occur :

1. The problem has a finite optimal value and $h(\bar{y}) \le y_0$ then $\bar{y}, y_0$ is an optimal solution of the original problem
2. The problem has a finite optimal value and $h(\bar{y}) > y_0$ then, we can use an optimal vector $\bar{u}$ of the sub-problem and generate a new constraint
$$\inf_{x \in X} \{f(x,y) + \bar{u}^T g(x,y) \le y_0\}$$
3. The sub-problem is not feasible, implying that $y \notin V$ and that one constraint
$$\inf_{x \in X} \{\lambda^T g(x,y)\} \le 0$$

is violated for a given λ. Add the violated constraint.

6.1 Pseudo-code

We give in algorithm 4

Algorithm 4 Generalized Benders Decomposition

```
Init : Let $\bar{y} \in Y \cap V$. Find an optimal solution $\bar{x}$ to
    the sub-problem
        $h(\bar{y}) = \text{minimize } f(x, \bar{y})$
                    $\text{s.t. } g(x, \bar{y}) \le 0$
                    $x \in X$
    as well as an optimal multiplier $\bar{u}$
    Assign $1 \to p$ and $0 \to q$ and define the first
    relaxed problem with constraint
        $\inf_{x \in X} \{f(x, y) + \bar{u}^T g(x, y)\} \le y_0$
    Also set $f(\bar{x}, \bar{y}) \to UB$ and a tolerance $\varepsilon$
Step 1 : Solve the relaxed problem
    $\min.\ y_0$
    $\text{s.t. } \inf_{x \in X} \{f(x, y) + {u^j}^T g(x, y)\} \le y_0 \quad j = 1...p$
    $\inf_{x \in X} \{{\lambda^j}^T g(x, y)\} \le 0 \quad j = 1...q$
    $y \in Y$
    Let $\bar{y}, \bar{y}_0$ be an optimal solution to the relaxed
    problem.
    If $UB - \bar{y}_0 \le \varepsilon$, STOP. The solution is $\varepsilon$-optimal
Step 2 : Solve the sub-problem sub-problem
        $h(\bar{y}) = \text{minimize } f(x, \bar{y})$
                    $\text{s.t. } g(x, \bar{y}) \le 0$
                    $x \in X$
    If the sub-problem is infeasible, find $\bar{\lambda} \in \Lambda$ such
        that
            $\inf_{x \in X} \{\bar{\lambda}^T g(x, y)\} > 0$
        Increment $q$ and introduce the constraint
            $\inf_{x \in X} \{\bar{\lambda}^T g(x, y)\} \le 0$
        in the relaxed problem then go to Step 1.
    Else if , there exists optimal solution to the
        sub-problem $\bar{x}$ such that $f(\bar{x}, \bar{y}) - \bar{y}_0 > \varepsilon$,
        then find an optimal multipler $\bar{u}$. Increment $p$ and add the following constraint to
        the relaxed problem :
            $\inf_{x \in X} \{f(x, y) + \bar{u}^T g(x, y)\} \le y_0$
        Set $\min(UB, f(\bar{x}, \bar{y})) \to UB$, go to Step 1.
    Else if , there exists optimal solution to the
        sub-problem $\bar{x}$ such that $f(\bar{x}, \bar{y}) - \bar{y}_0 \le \varepsilon$,
        then STOP. $\bar{x}, \bar{y}$ is a $\varepsilon$-optimal.
```

6.2 Examples

Toy bilinear problem

Two-stage Stochastic problme with recourse

Chapter 11

Dantzig-Wolfe decomposition

Appendices

Appendix A

Linear algebra

Appendix B

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 $x_1, ..., x_n$ be a finite set of vectors in a real vector space, a convex combination of these vectors is a vector of the form*

$$\sum_{i=1}^{n} \alpha_i x_i \qquad \textit{with} \ \sum_{i=1}^{k} \alpha_i = 1, \alpha \geq 0$$

Definition 5 (Convex hull)**.**

$$\textit{conv}(X) = \left\{ \sum_{i=1}^{n} \alpha_i x_i \middle| x_i \in X, \sum_{i=1}^{n} \alpha_i = 1, \alpha \geq 0 \right\}$$

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

$$\sum_{i=1}^{k} \alpha_i x_i \qquad \textit{with} \ \alpha \geq 0$$

Definition 7 (Conical hull)**.**

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

Polyhedra, polytopes and polyhedral cones

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

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

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

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

Definition 10 (Polyhedral cone)**.**

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

Observation 3 (Important)**.** *Note that no real consensus has been reach on the definitions of polyhedra and polytopes. Sometimes, polyhedra corresponds to three dimensional solids with polygonial 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 B.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 Minkowki sum is defined as*

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

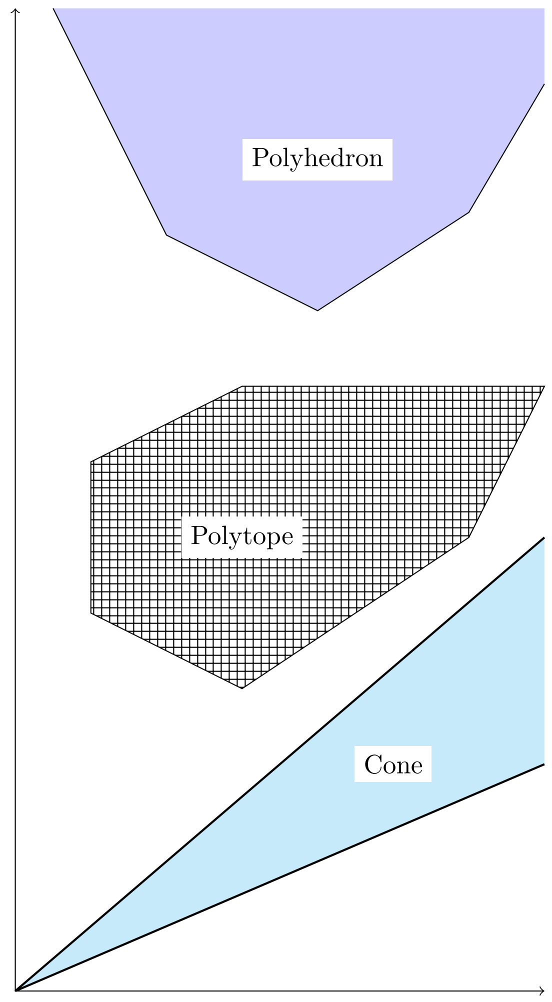


Figure B.1: A polyhdron, a polytope and a polyhedral cone depicted in 2D

Theorem 8 (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* $x_1, ..., x_q \in \mathbb{R}^{\cdot}n$ *and* $y_1, ..., y_r \in \mathbb{R}^n$ *such that*

$$P = cone(x_1, ..., x_q) + conv(y_1, ..., y_r)$$

Theorem 9 (Decomposition theorem for polyhedra)**.** *A set* P *of vectors in a Euclidean space is a polyhedron if and only if it is the Minkowki sum of a polytope* Q *and a polyhedral cone* C*.*

$$P = Q + C$$

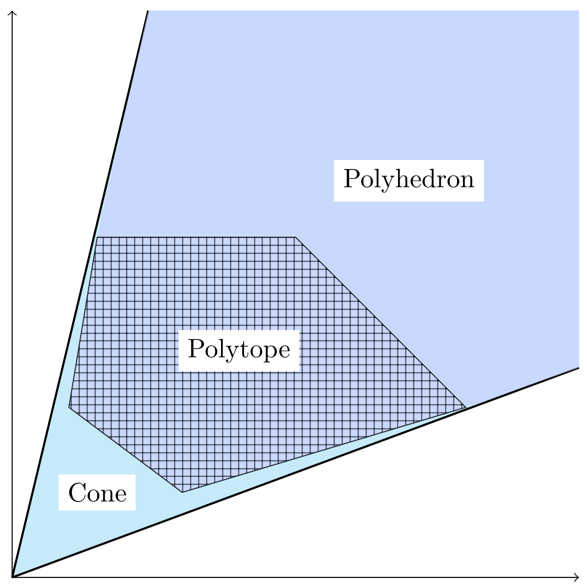


Figure B.2: Illustration of theorem 9

This decomposition theorem is illustrated in figure B.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] A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, Oct 1972.

[3] E. Gol'shtein. *Linear programming, by D.B.Yudin and E.G.Gol'shtein.* publisher not identified.

[4] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming.* Addison-Wesley, 1973.

List of Figures

List of Tables

List of Algorithms