

Introduction à l'Optimisation Numérique

Quentin Louveaux

ULg - Institut Montefiore

2009

Chapter 3

The simplex method

The simplex algorithm

Principle

- Start at a **feasible basic solution**
- Check whether the current basis is **optimal**
- If not, find a **direction of improvement**
- The direction of improvement leads to either ... a better **feasible basic solution**
- or ... proving that the problem **is unbounded**

Fundamental question

Let x be a feasible basic solution.

Find a **direction d** such that $x + \theta d$ is feasible for some $\theta \geq 0$

The simplex algorithm

Principle

- Start at a **feasible basic solution**
- Check whether the current basis is **optimal**
- If not, find a **direction of improvement**
- The direction of improvement leads to either ... a better **feasible basic solution**
- or ... proving that the problem **is unbounded**

Fundamental question

Let x be a feasible basic solution.

Find a **direction** d such that $x + \theta d$ is feasible for some $\theta \geq 0$

The simplex algorithm

Principle

- Start at a **feasible basic solution**
- Check whether the current basis is **optimal**
- If not, find a **direction of improvement**
- The direction of improvement leads to either ... a better **feasible basic solution**
- or ... proving that the problem **is unbounded**

Fundamental question

Let x be a feasible basic solution.

Find a **direction d** such that $x + \theta d$ is feasible for some $\theta \geq 0$

Finding a feasible direction from a basic feasible solution

Given : $x = (x_B \ x_N) \in P$

Find d such that $x + d\theta \in P$

- We want to change at least one **nonbasic** solution.

Select **one index $j \in N$**

$$d_j = 1 \quad d_i = 0 \text{ for all } i \neq j$$

- We have $Ax = b$ and $A(x + \theta d) = b \Rightarrow Ad = 0$

-

$$0 = Ad = \sum_{i=1}^n A_{B(i)} d_{B(i)} + A_j d_j$$
$$Bd_B + A_j d_j$$

- $d_B = -B^{-1}A_j$ is the j^{th} basic direction

Finding a feasible direction from a basic feasible solution

Given : $x = (x_B \ x_N) \in P$

Find d such that $x + d\theta \in P$

- We want to change at least one **nonbasic** solution.

Select **one index** $j \in N$

$$d_j = 1 \quad d_i = 0 \text{ for all } i \neq j$$

- We have $Ax = b$ and $A(x + \theta d) = b \Rightarrow Ad = 0$

•

$$0 = Ad = \sum_{i=1}^n A_{B(i)} d_{B(i)} + A_j d_j$$
$$Bd_B + A_j d_j$$

- $d_B = -B^{-1}A_j$ is the j^{th} basic direction

Finding a feasible direction from a basic feasible solution

Given : $x = (x_B \ x_N) \in P$

Find d such that $x + d\theta \in P$

- We want to change at least one **nonbasic** solution.

Select **one index** $j \in N$

$$d_j = 1 \quad d_i = 0 \text{ for all } i \neq j$$

- We have $Ax = b$ and $A(x + \theta d) = b \Rightarrow Ad = 0$

-

$$0 = Ad = \sum_{i=1}^n A_{B(i)} d_{B(i)} + A_j d_j$$
$$Bd_B + A_j d_j = 0$$

- $d_B = -B^{-1}A_j$ is the j^{th} basic direction

Finding a feasible direction from a basic feasible solution

Given : $x = (x_B \ x_N) \in P$

Find d such that $x + d\theta \in P$

- We want to change at least one **nonbasic** solution.

Select **one index $j \in N$**

$$d_j = 1 \quad d_i = 0 \text{ for all } i \neq j$$

- We have $Ax = b$ and $A(x + \theta d) = b \Rightarrow Ad = 0$

-

$$0 = Ad = \sum_{i=1}^n A_{B(i)} d_{B(i)} + A_j d_j$$
$$Bd_B + A_j d_j = 0$$

- $d_B = -B^{-1}A_j$ is the j^{th} **basic direction**

Taking care of the nonnegativity constraints

- **Nonbasic variables** : $x_i (i \neq j) := 0 \Rightarrow$ OK!
 x_j : goes in the positive direction \Rightarrow OK!
- **Basic Variables**
Nondegenerate case : $x_B > 0$
Therefore $x_B + \theta d_B \geq 0$ for θ sufficiently small
- **Degenerate case** : $x_{B(i)} = 0$ for some i
2 cases
(1) $(d_B)_i = (-B^{-1}A_j)_i \geq 0 \Rightarrow$ feasible for θ small enough
(2) $(d_B)_i = (-B^{-1}A_j)_i < 0 \Rightarrow$ not feasible for any θ

Taking care of the nonnegativity constraints

- **Nonbasic variables** : $x_i (i \neq j) := 0 \Rightarrow$ OK!
 x_j : goes in the positive direction \Rightarrow OK!
- **Basic Variables**
Nondegenerate case : $x_B > 0$
Therefore $x_B + \theta d_B \geq 0$ for θ **sufficiently small**
- **Degenerate case** : $x_{B(i)} = 0$ for some i
2 cases
(1) $(d_B)_i = (-B^{-1}A_j)_i \geq 0 \Rightarrow$ feasible for θ small enough
(2) $(d_B)_i = (-B^{-1}A_j)_i < 0 \Rightarrow$ not feasible for any θ

Taking care of the nonnegativity constraints

- **Nonbasic variables** : $x_i (i \neq j) := 0 \Rightarrow$ OK!
 x_j : goes in the positive direction \Rightarrow OK!
- **Basic Variables**
Nondegenerate case : $x_B > 0$
Therefore $x_B + \theta d_B \geq 0$ for θ **sufficiently small**
- **Degenerate case** : $x_{B(i)} = 0$ for some i
2 cases
(1) $(d_B)_i = (-B^{-1}A_j)_i \geq 0 \Rightarrow$ feasible for θ small enough
(2) $(d_B)_i = (-B^{-1}A_j)_i < 0 \Rightarrow$ not feasible for any θ

Checking optimality

Essential question : Is the current basis **optimal** or can we find a **basic direction** that improves the objective function ?

Definition : $c_B = (c_{B(1)}, \dots, c_{B(m)})$

Let us compute the **rate of objective change** for the j^{th} basic direction.

$d = (d_B \ d_N)$ with $d_B = -B^{-1}A_j$ $d_N = e_j$

$$\begin{aligned}c^T d &= c^T d_B + c^T d_N \\ &= -c_B^T B^{-1} A_j + c_j\end{aligned}$$

Definition

The **reduced cost** of the j^{th} **nonbasic variable** is

$$\bar{c}_j = c_j - c_B^T B^{-1} A_j.$$

Checking optimality

Essential question : Is the current basis **optimal** or can we find a **basic direction** that improves the objective function ?

Definition : $c_B = (c_{B(1)}, \dots, c_{B(m)})$

Let us compute the **rate of objective change** for the j^{th} basic direction.

$d = (d_B \ d_N)$ with $d_B = -B^{-1}A_j$ $d_N = e_j$

$$\begin{aligned}c^T d &= c^T d_B + c^T d_N \\ &= -c_B^T B^{-1} A_j + c_j\end{aligned}$$

Definition

The **reduced cost** of the j^{th} **nonbasic variable** is

$$\bar{c}_j = c_j - c_B^T B^{-1} A_j.$$

Optimality Criterion

Theorem

Consider a basic feasible solution x

- (i) If $\bar{c} \geq 0$ then x is **optimal**
- (ii) If x is optimal and nondegenerate then $\bar{c} \geq 0$.

Definition

A basis matrix B is said to be **optimal** if

- (i) $B^{-1}b \geq 0$ and
- (ii) $\bar{c} \equiv -c_B^T B^{-1}A \geq 0$

Optimality Criterion

Theorem

Consider a basic feasible solution x

- (i) If $\bar{c} \geq 0$ then x is **optimal**
- (ii) If x is optimal and nondegenerate then $\bar{c} \geq 0$.

Definition

A basis matrix B is said to be **optimal** if

- (i) $B^{-1}b \geq 0$ and
- (ii) $\bar{c} = -c_B^T B^{-1}A \geq 0$

Simplex algorithm

Iterative step : From a basis B do

If $\bar{c}_j \geq 0$ for all j

then the current basis is optimal

else select a nonbasic variable j with $\bar{c}_j < 0$

We bring j into the basis

We now look for θ maximal such that $x + \theta d \in P$

If $d_i \geq 0$ for all $i \in B$

then $x + \theta d \geq 0$ for all $\theta \geq 0$

*The problem is **unbounded** and $OPT = -\infty$*

If $d_i < 0$ for some $i \in B$

then $\theta^* = \min_{\{i \in B \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right)$.

Variable i achieving the minimum goes out of the basis

We move to the next vertex (basic feasible solution)

$B \leftarrow B \cup \{j\} \setminus \{i\}$

New point := $x + \theta^* d$

Simplex algorithm

Iterative step : From a basis B do

If $\bar{c}_j \geq 0$ for all j

then the current basis is optimal

else select a nonbasic variable j with $\bar{c}_j < 0$

We bring j into the basis

We now look for θ maximal such that $x + \theta d \in P$

If $d_i \geq 0$ for all $i \in B$

then $x + \theta d \geq 0$ for all $\theta \geq 0$

*The problem is **unbounded** and $OPT = -\infty$*

If $d_i < 0$ for some $i \in B$

then $\theta^* = \min_{\{i \in B \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right)$.

Variable i achieving the minimum goes out of the basis

We move to the next vertex (basic feasible solution)

$B \leftarrow B \cup \{j\} \setminus \{i\}$

New point := $x + \theta^* d$

Simplex algorithm

Iterative step : From a basis B do

If $\bar{c}_j \geq 0$ for all j

then the current basis is optimal

else select a nonbasic variable j with $\bar{c}_j < 0$

We bring j into the basis

We now look for θ maximal such that $x + \theta d \in P$

If $d_i \geq 0$ for all $i \in B$

then $x + \theta d \geq 0$ for all $\theta \geq 0$

*The problem is **unbounded** and $OPT = -\infty$*

If $d_i < 0$ for some $i \in B$

then $\theta^* = \min_{\{i \in B \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right)$.

Variable i achieving the minimum goes out of the basis

We move to the next vertex (basic feasible solution)

$B \leftarrow B \cup \{j\} \setminus \{i\}$

New point := $x + \theta^* d$

Simplex algorithm

Iterative step : From a basis B do

If $\bar{c}_j \geq 0$ for all j

then the current basis is optimal

else select a nonbasic variable j with $\bar{c}_j < 0$

We bring j into the basis

We now look for θ maximal such that $x + \theta d \in P$

If $d_i \geq 0$ for all $i \in B$

then $x + \theta d \geq 0$ for all $\theta \geq 0$

*The problem is **unbounded** and $OPT = -\infty$*

If $d_i < 0$ for some $i \in B$

then $\theta^* = \min_{\{i \in B \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right)$.

Variable i achieving the minimum goes out of the basis

We move to the next vertex (basic feasible solution)

$B \leftarrow B \cup \{j\} \setminus \{i\}$

New point := $x + \theta^* d$

Simplex algorithm

Iterative step : From a basis B do

If $\bar{c}_j \geq 0$ for all j

then the current basis is optimal

else select a nonbasic variable j with $\bar{c}_j < 0$

We bring j into the basis

We now look for θ maximal such that $x + \theta d \in P$

If $d_i \geq 0$ for all $i \in B$

then $x + \theta d \geq 0$ for all $\theta \geq 0$

*The problem is **unbounded** and $OPT = -\infty$*

If $d_i < 0$ for some $i \in B$

then $\theta^* = \min_{\{i \in B \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right)$.

Variable i achieving the minimum goes out of the basis

We move to the next vertex (basic feasible solution)

$B \leftarrow B \cup \{j\} \setminus \{i\}$

New point := $x + \theta^* d$

Simplex algorithm

Iterative step : From a basis B do

If $\bar{c}_j \geq 0$ for all j

then the current basis is optimal

else select a nonbasic variable j with $\bar{c}_j < 0$

We bring j into the basis

We now look for θ maximal such that $x + \theta d \in P$

If $d_i \geq 0$ for all $i \in B$

then $x + \theta d \geq 0$ for all $\theta \geq 0$

*The problem is **unbounded** and $OPT = -\infty$*

If $d_i < 0$ for some $i \in B$

then $\theta^* = \min_{\{i \in B \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right)$.

Variable i achieving the minimum goes out of the basis

We move to the next vertex (basic feasible solution)

$$B \leftarrow B \cup \{j\} \setminus \{i\}$$

$$\text{New point} := x + \theta^* d$$

Simplex algorithm

Iterative step : From a basis B do

If $\bar{c}_j \geq 0$ for all j

then the current basis is optimal

else select a nonbasic variable j with $\bar{c}_j < 0$

We bring j into the basis

We now look for θ maximal such that $x + \theta d \in P$

If $d_i \geq 0$ for all $i \in B$

then $x + \theta d \geq 0$ for all $\theta \geq 0$

*The problem is **unbounded** and $OPT = -\infty$*

If $d_i < 0$ for some $i \in B$

then $\theta^* = \min_{\{i \in B \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right)$.

Variable i achieving the minimum goes out of the basis

We move to the next vertex (basic feasible solution)

$$B \leftarrow B \cup \{j\} \setminus \{i\}$$

$$\text{New point} := x + \theta^* d$$

The operation of moving to another basis

In an iteration of the simplex algorithm, in the basis, we replace the variable i such that

$$i = \arg \min \left\{ -\frac{x_{B(i)}}{d_{B(i)}} \mid i \in B \text{ with } d_{B(i)} < 0 \right\}$$

by the variable j **entering the basis** (that was chosen with $\bar{c}_j < 0$).

Theorem

- (i) The columns $A_{B(k)}$, $k \neq i$ and A_j are linearly independent
- (ii) The vector $y = x + \theta^* d$ is a basic feasible solution associated with the new basis matrix.

The operation of moving to another basis

In an iteration of the simplex algorithm, in the basis, we replace the variable i such that

$$i = \arg \min \left\{ -\frac{x_{B(i)}}{d_{B(i)}} \mid i \in B \text{ with } d_{B(i)} < 0 \right\}$$

by the variable j **entering the basis** (that was chosen with $\bar{c}_j < 0$).

Theorem

- (i) The columns $A_{B(k)}$, $k \neq i$ and A_j are linearly independent
- (ii) The vector $y = x + \theta^* d$ is a basic feasible solution associated with the new basis matrix.

Finiteness of the simplex algorithm

Theorem

If **all bases of P are nondegenerate** then the simplex algorithm terminates in a finite number of iterations. At termination, there are two possibilities

- We have an optimal basis and an associated basic feasible solution which is optimal
- We have found a vector d such that $Ad = 0$ and $c^T d < 0$ implying that the optimal cost is $-\infty$

The degenerate case

We may do a pivot (j enters the basis and i goes out) and **not changing the corresponding basic feasible solution**. This is called a **degenerate pivot**. Using degenerate pivots, two possibilities occur

- After some degenerate pivots, we find a **direction of improvement** and move to a basis with a **different corresponding solution**
- After some degenerate pivots, we come back to the **same basis** \rightarrow the algorithm **cycles**.
- For a good **pivot rule**, we can avoid cycling (see next section).

Finiteness of the simplex algorithm

Theorem

If **all bases of P are nondegenerate** then the simplex algorithm terminates in a finite number of iterations. At termination, there are two possibilities

- We have an optimal basis and an associated basic feasible solution which is optimal
- We have found a vector d such that $Ad = 0$ and $c^T d < 0$ implying that the optimal cost is $-\infty$

The degenerate case

We may do a pivot (j enters the basis and i goes out) and **not changing the corresponding basic feasible solution**. This is called a **degenerate pivot**. Using degenerate pivots, two possibilities occur

- After some degenerate pivots, we find a **direction of improvement** and move to a basis with a **different corresponding solution**
- After some degenerate pivots, we come back to the **same basis** \rightarrow the algorithm **cycles**.
- For a good **pivot rule**, we can avoid cycling (see next section).

Implementation issues

Implementing a good linear software is not a trivial task.

We will see that the way the simplex method is implemented is not the most obvious one.

Full Tableau Implementation

We do not compute the inverse matrix B^{-1} but maintain at each iteration the **simplex tableau**

$$B^{-1}(b \mid A).$$

A pivot = **elementary row operations**

Complexity :

Memory $\mathcal{O}(mn)$

Number of operations $\mathcal{O}(mn)$.

Drawback : possibility of **fill in**

Implementation issues

The revised simplex

Update the **inverse matrix** B^{-1} instead of the **full tableau** at each iteration.

- (i) Start with a basis $A_{B(1)}, \dots, A_{B(m)}$ and the inverse B^{-1}
- (ii) Compute the **reduced costs** $\bar{c}_j = c_j - c_B^T B^{-1} A_j$.
Choose an index j with $\bar{c}_j < 0$ (if none exists, return **optimal**).
- (iii) Compute $u = B^{-1} A_j$
- (iv) Compute

$$\theta^* = \min_{i=1, \dots, m | u_i > 0} \frac{x_{B(i)}}{u_i}.$$

- (v) Let l be the index achieving the minimum, the new basic solution is

$$y_j = \theta^*, y_{B(i)} = x_{B(i)} - \theta^* u_i, i \neq l.$$

- (vi) Perform the corresponding row operations on B^{-1} in order to compute the **new inverse matrix**.

Complexity :

Memory : $\mathcal{O}(m^2)$ (because A is sparse most of the times)

Operations : $\mathcal{O}(m^2)$ for the update of the inverse and $\mathcal{O}(mn)$ for the computation of the reduced cost \rightarrow this can be reduced if we find a good variable quickly !

Implementation issues

The revised simplex

Update the **inverse matrix** B^{-1} instead of the **full tableau** at each iteration.

- (i) Start with a basis $A_{B(1)}, \dots, A_{B(m)}$ and the inverse B^{-1}
- (ii) Compute the **reduced costs** $\bar{c}_j = c_j - c_B^T B^{-1} A_j$.
Choose an index j with $\bar{c}_j < 0$ (if none exists, return **optimal**).
- (iii) Compute $u = B^{-1} A_j$
- (iv) Compute

$$\theta^* = \min_{i=1, \dots, m | u_i > 0} \frac{x_{B(i)}}{u_i}.$$

- (v) Let l be the index achieving the minimum, the new basic solution is

$$y_j = \theta^*, y_{B(i)} = x_{B(i)} - \theta^* u_i, i \neq l.$$

- (vi) Perform the corresponding row operations on B^{-1} in order to compute the **new inverse matrix**.

Complexity :

Memory : $\mathcal{O}(m^2)$ (because A is sparse most of the times)

Operations : $\mathcal{O}(m^2)$ for the update of the inverse and $\mathcal{O}(mn)$ for the computation of the reduced cost \rightarrow this can be reduced if we find a good variable quickly !

Implementation issues

The revised simplex

Update the **inverse matrix** B^{-1} instead of the **full tableau** at each iteration.

- (i) Start with a basis $A_{B(1)}, \dots, A_{B(m)}$ and the inverse B^{-1}
- (ii) Compute the **reduced costs** $\bar{c}_j = c_j - c_B^T B^{-1} A_j$.
Choose an index j with $\bar{c}_j < 0$ (if none exists, return **optimal**).
- (iii) Compute $u = B^{-1} A_j$
- (iv) Compute

$$\theta^* = \min_{i=1, \dots, m | u_i > 0} \frac{x_{B(i)}}{u_i}.$$

- (v) Let l be the index achieving the minimum, the new basic solution is

$$y_j = \theta^*, y_{B(i)} = x_{B(i)} - \theta^* u_i, i \neq l.$$

- (vi) Perform the corresponding row operations on B^{-1} in order to compute the **new inverse matrix**.

Complexity :

Memory : $\mathcal{O}(m^2)$ (because A is sparse most of the times)

Operations : $\mathcal{O}(m^2)$ for the update of the inverse and $\mathcal{O}(mn)$ for the computation of the reduced cost \rightarrow this can be reduced if we find a good variable quickly !

Implementation issues

The revised simplex

Update the **inverse matrix** B^{-1} instead of the **full tableau** at each iteration.

- (i) Start with a basis $A_{B(1)}, \dots, A_{B(m)}$ and the inverse B^{-1}
- (ii) Compute the **reduced costs** $\bar{c}_j = c_j - c_B^T B^{-1} A_j$.
Choose an index j with $\bar{c}_j < 0$ (if none exists, return **optimal**).
- (iii) Compute $u = B^{-1} A_j$
- (iv) Compute

$$\theta^* = \min_{i=1, \dots, m | u_i > 0} \frac{x_{B(i)}}{u_i}.$$

- (v) Let l be the index achieving the minimum, the new basic solution is

$$y_j = \theta^*, y_{B(i)} = x_{B(i)} - \theta^* u_i, i \neq l.$$

- (vi) Perform the corresponding row operations on B^{-1} in order to compute the **new inverse matrix**.

Complexity :

Memory : $\mathcal{O}(m^2)$ (because A is sparse most of the times)

Operations : $\mathcal{O}(m^2)$ for the update of the inverse and $\mathcal{O}(mn)$ for the computation of the reduced cost \rightarrow this can be reduced if we find a good variable quickly !

Implementation issues

The revised simplex

Update the **inverse matrix** B^{-1} instead of the **full tableau** at each iteration.

- (i) Start with a basis $A_{B(1)}, \dots, A_{B(m)}$ and the inverse B^{-1}
- (ii) Compute the **reduced costs** $\bar{c}_j = c_j - c_B^T B^{-1} A_j$.
Choose an index j with $\bar{c}_j < 0$ (if none exists, return **optimal**).
- (iii) Compute $u = B^{-1} A_j$
- (iv) Compute

$$\theta^* = \min_{i=1, \dots, m | u_i > 0} \frac{x_{B(i)}}{u_i}.$$

- (v) Let l be the index achieving the minimum, the new basic solution is

$$y_j = \theta^*, y_{B(i)} = x_{B(i)} - \theta^* u_i, i \neq l.$$

- (vi) Perform the corresponding row operations on B^{-1} in order to compute the **new inverse matrix**.

Complexity :

Memory : $\mathcal{O}(m^2)$ (because A is sparse most of the times)

Operations : $\mathcal{O}(m^2)$ for the update of the inverse and $\mathcal{O}(mn)$ for the computation of the reduced cost \rightarrow this can be reduced if we find a good variable quickly !

Implementation issues

The revised simplex

Update the **inverse matrix** B^{-1} instead of the **full tableau** at each iteration.

- (i) Start with a basis $A_{B(1)}, \dots, A_{B(m)}$ and the inverse B^{-1}
- (ii) Compute the **reduced costs** $\bar{c}_j = c_j - c_B^T B^{-1} A_j$.
Choose an index j with $\bar{c}_j < 0$ (if none exists, return **optimal**).
- (iii) Compute $u = B^{-1} A_j$
- (iv) Compute

$$\theta^* = \min_{i=1, \dots, m | u_i > 0} \frac{x_{B(i)}}{u_i}.$$

- (v) Let l be the index achieving the minimum, the new basic solution is

$$y_j = \theta^*, y_{B(i)} = x_{B(i)} - \theta^* u_i, i \neq l.$$

- (vi) Perform the corresponding row operations on B^{-1} in order to compute the **new inverse matrix**.

Complexity :

Memory : $\mathcal{O}(m^2)$ (because A is sparse most of the times)

Operations : $\mathcal{O}(m^2)$ for the update of the inverse and $\mathcal{O}(mn)$ for the computation of the reduced cost \rightarrow this can be reduced if we find a good variable quickly !

Implementation issues

The revised simplex

Update the **inverse matrix** B^{-1} instead of the **full tableau** at each iteration.

- (i) Start with a basis $A_{B(1)}, \dots, A_{B(m)}$ and the inverse B^{-1}
- (ii) Compute the **reduced costs** $\bar{c}_j = c_j - c_B^T B^{-1} A_j$.
Choose an index j with $\bar{c}_j < 0$ (if none exists, return **optimal**).
- (iii) Compute $u = B^{-1} A_j$
- (iv) Compute

$$\theta^* = \min_{i=1, \dots, m | u_i > 0} \frac{x_{B(i)}}{u_i}.$$

- (v) Let l be the index achieving the minimum, the new basic solution is

$$y_j = \theta^*, y_{B(i)} = x_{B(i)} - \theta^* u_i, i \neq l.$$

- (vi) Perform the corresponding row operations on B^{-1} in order to compute the **new inverse matrix**.

Complexity :

Memory : $\mathcal{O}(m^2)$ (because A is sparse most of the times)

Operations : $\mathcal{O}(m^2)$ for the update of the inverse and $\mathcal{O}(mn)$ for the computation of the reduced cost \rightarrow this can be reduced if we find a good variable quickly !

Anticycling rules

Lexicographic order

We define a **total ordering** on vectors of \mathbb{R}^n . We say that $a \in \mathbb{R}^n$ is **lexicographically smaller** than $b \in \mathbb{R}^n$, $a \prec^L b$ if **the first nonzero component of $b - a$ is positive**.

Example

$$(1; 2 \ 3 \ 4) \prec^L (1 \ 3; 0 \ 0)$$

$$(3; 1; -2; 7) \prec^L (7 \ 0 \ -3; 1)$$

$$(1; 2 \ 3) \prec^L (1 \ 2 \ 7)$$

Lexicographic pivoting rule

- (i) Choose a variable x_i ($i \in N$) with $\bar{c}_i < 0$ arbitrarily
- (ii) For each i with $\frac{x_B(i)}{u_i} (u_i > 0)$ minimum select the variable with the corresponding tableau row which is lexicographically minimal to leave the basis.

Anticycling rules

Lexicographic order

We define a **total ordering** on vectors of \mathbb{R}^n . We say that $a \in \mathbb{R}^n$ is **lexicographically smaller** than $b \in \mathbb{R}^n$, $a \prec^L b$ if **the first nonzero component of $b - a$ is positive**.

Example

$$(1; 2 \ 3 \ 4) \prec^L (1 \ 3; 0 \ 0)$$

$$(3; 1; -2; 7) \prec^L (7 \ 0 \ -3; 1)$$

$$(1; 2 \ 3) \prec^L (1 \ 2 \ 7)$$

Lexicographic pivoting rule

- (i) Choose a variable x_i ($i \in N$) with $\bar{c}_i < 0$ **arbitrarily**
- (ii) For each i with $\frac{x_{B(i)}}{u_i} (u_i > 0)$ minimum select the variable with the corresponding **tableau row** which is **lexicographically minimal** to leave the basis.

Lexicographic pivoting rule

In the tableau form, we use the following convention

- the 0^{th} row corresponds to the **reduced costs**
- the 0^{th} column corresponds to the **right-hand-sides**
- the element in the 0^{th} row and 0^{th} column corresponds to the opposite of the objective.

Theorem

Suppose that all rows (except the 0^{th} row) start being **lexicographically positive**. If we use the lexicographic pivoting rule then

- Every row (except the 0^{th} row) remain **lexicographically positive** during the simplex algorithm
- The zeroth row **strictly increases lexicographically** during the simplex algorithm
- The simplex algorithm terminates after a finite number of steps.

Lexicographic pivoting rule

In the tableau form, we use the following convention

- the 0^{th} row corresponds to the **reduced costs**
- the 0^{th} column corresponds to the **right-hand-sides**
- the element in the 0^{th} row and 0^{th} column corresponds to the opposite of the objective.

Theorem

Suppose that all rows (except the 0^{th} row) start being **lexicographically positive**. If we use the lexicographic pivoting rule then

- Every row (except the 0^{th} row) remain **lexicographically positive** during the simplex algorithm
- The zeroth row **strictly increases lexicographically** during the simplex algorithm
- The simplex algorithm terminates after a finite number of steps.

Finite pivoting rule

The following pivoting rule is extremely simple to use with the revised simplex algorithm

Bland's rule

- (i) The variable i that enters the basis corresponds to the **smallest subscript** such that $\bar{c}_i < 0$
- (ii) If there is a tie between variables to exit the basis, select the one **with the smallest subscript**

Using Bland's rule, the simplex algorithm is guaranteed to terminate after a finite number of steps.

Finding an initial basis

2 issues : finding a **feasible solution** and finding **the corresponding basis**.

One easy case

Consider a problem of the type $Ax \leq b$ where $b \geq 0$.

The **standard form**, if we add the **slack variables**

$$\begin{array}{rcl} a_{11}x_1 + \cdots + a_{1n}x_n + s_1 & = & b_1 \\ & \vdots & \\ a_{m1}x_1 + \cdots + a_{mn}x_n & + s_m = & b_m \end{array}$$

The point $(x_1 = 0, \dots, x_n = 0)$ is a feasible solution and $(s_1 \dots s_m)$ is the corresponding basis.

In general

It is not easy to find a **feasible basis**.

One can try any basis (choice of m variables) but there is **no guarantee that it is feasible**. In some cases, the problem is **infeasible** \rightarrow it might be impossible to find a basic feasible solution.

Finding an initial basis

2 issues : finding a **feasible solution** and finding **the corresponding basis**.

One easy case

Consider a problem of the type $Ax \leq b$ where $b \geq 0$.

The **standard form**, if we add the **slack variables**

$$\begin{array}{rcl} a_{11}x_1 + \cdots + a_{1n}x_n + s_1 & = & b_1 \\ & \vdots & \\ & \vdots & \\ a_{m1}x_1 + \cdots + a_{mn}x_n & + s_m = & b_m \end{array}$$

The point $(x_1 = 0, \dots, x_n = 0)$ is a feasible solution and $(s_1 \dots s_m)$ is the corresponding basis.

In general

It is not easy to find a **feasible basis**.

One can try any basis (choice of m variables) but there is **no guarantee that it is feasible**. In some cases, the problem is **infeasible** \rightarrow it might be impossible to find a basic feasible solution.

Using a linear program and the simplex algorithm to find an initial basis

Phase I

Consider a problem of the type $Ax = b, x \in \mathbb{R}^n$ where we assume (after suitable multiplication of rows by -1) that $b \geq 0$.

$$\begin{array}{llll} \min & & \xi_1 + \cdots + & \xi_m \\ & a_{11}x_1 + \cdots + a_{1n}x_n & + \xi_1 & = b_1 \\ & & \vdots & \vdots \\ & a_{m1}x_1 + \cdots + a_{mn}x_n & + \xi_m & = b_m \end{array}$$

- If the optimal solution of the phase 1 problem is $> 0 \Rightarrow$ the initial problem is infeasible.
- If the initial problem is feasible \Rightarrow the phase 1 problem has 0 as optimal solution
- One more issue : how to obtain an initial basis for the original problem without ξ_i variables in it

Using a linear program and the simplex algorithm to find an initial basis

Phase I

Consider a problem of the type $Ax = b, x \in \mathbb{R}^n$ where we assume (after suitable multiplication of rows by -1) that $b \geq 0$.

$$\begin{array}{rcl} \min & \xi_1 + \cdots + \xi_m & \\ & a_{11}x_1 + \cdots + a_{1n}x_n + \xi_1 & = b_1 \\ & \vdots & \vdots \\ & a_{m1}x_1 + \cdots + a_{mn}x_n + \xi_m & = b_m \end{array}$$

- If the optimal solution of the phase 1 problem is $> 0 \Rightarrow$ the initial problem is infeasible.
- If the initial problem is feasible \Rightarrow the phase 1 problem has 0 as optimal solution
- One more issue : how to obtain an initial basis for the original problem without ξ_i variables in it

Using a linear program and the simplex algorithm to find an initial basis

Phase I

Consider a problem of the type $Ax = b, x \in \mathbb{R}^n$ where we assume (after suitable multiplication of rows by -1) that $b \geq 0$.

$$\begin{array}{rcl} \min & \xi_1 + \cdots + \xi_m & \\ & a_{11}x_1 + \cdots + a_{1n}x_n + \xi_1 & = b_1 \\ & \vdots & \vdots \\ & a_{m1}x_1 + \cdots + a_{mn}x_n + \xi_m & = b_m \end{array}$$

- If the optimal solution of the phase 1 problem is $> 0 \Rightarrow$ the initial problem is infeasible.
- If the initial problem is feasible \Rightarrow the phase 1 problem has 0 as optimal solution
- One more issue : how to obtain an initial basis for the original problem without ξ_i variables in it

The big-M method

Phase I and Phase II together

Consider a problem of the type $Ax = b, x \in \mathbb{R}_+^n$ where we assume (after suitable multiplication of rows by -1) that $b \geq 0$.

$$\begin{array}{llll} \min & c_1x_1 + \cdots + c_nx_n & + M\xi_1 + \cdots + M\xi_m & \\ & a_{11}x_1 + \cdots + a_{1n}x_n & + \xi_1 & = b_1 \\ & & \vdots & \vdots \\ & a_{m1}x_1 + \cdots + a_{mn}x_n & + \xi_m & = b_m \end{array}$$

- M is always considered as very big
- When an auxiliary variable ξ_i exits the basis, we can get rid of the full column.

The big-M method

Phase I and Phase II together

Consider a problem of the type $Ax = b, x \in \mathbb{R}_+^n$ where we assume (after suitable multiplication of rows by -1) that $b \geq 0$.

$$\begin{array}{llll} \min & c_1x_1 + \cdots + c_nx_n & + M\xi_1 + \cdots + M\xi_m & \\ & a_{11}x_1 + \cdots + a_{1n}x_n & + \xi_1 & = b_1 \\ & & \vdots & \vdots \\ & a_{m1}x_1 + \cdots + a_{mn}x_n & + \xi_m & = b_m \end{array}$$

- M is always considered as very big
- When an auxiliary variable ξ_i exits the basis, we can get rid of the full column.