Optimisation

Lecture 4 - The Simplex Method

Fall semester - 2024

Dr. Eng. Valentin Leplat

Innopolis University

September 17, 2024

Outline

Introduction

Simplex method (or simplex algorithm)

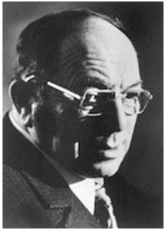

Figure: Leonid KANTOROVICH (1912-1986)

- Kantorovich worked for the Soviet government. He was given the task of optimizing production in industry.
- He proposed in 1939 the mathematical technique now known as linear programming, some years before it was reinvented and much advanced by **George Dantzig** (1914-2005).
- He was awarded the 1975 Nobel Laureate in Economics for 'contributions to the theory of optimum allocation of resources'.

Simplex method (or simplex algorithm)

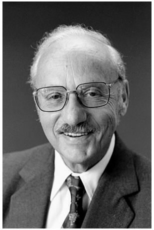

Figure: George Dantzig (1914-2005)

- 1947: George Dantzig, at the RAND Corporation, creates the simplex method for linear programming.
- **In terms of widespread application, Dantzig's algorithm is one of the most successful of all time**: Linear programming dominates the world of industry, where economic survival depends on the ability to optimize within budgetary and other constraints.[1]
- The simplex method is an elegant way of arriving at optimal answers. Although theoretically susceptible to exponential delays, the algorithm in practice is highly efficient-which in itself says something interesting about the nature of computation.

Ref. The Best of the 20th Century: Editors Name Top 10 Algorithms, Link, A. Cipra.

[1] of course, the real problems of industry are often nonlinear; the use of linear programming is sometimes dictated by the computational budget.

Recall: The fundamental Theorem

$$\min_x \quad c^T x \quad \text{such that} \quad Ax \geqslant b.$$

The set $\mathcal{P} = \{x \mid Ax \geqslant b\}$ is a Polyhedron.

Recall: The fundamental Theorem

$$\min_x \quad c^T x \quad \text{such that} \quad Ax \geqslant b.$$

The set $\mathcal{P} = \{x \mid Ax \geqslant b\}$ is a Polyhedron.

Fundamental theorem.

If a linear optimization problem has finite optimal cost and the polyhedron $\mathcal{P}$ has a vertex, then there is a vertex of $\mathcal{P}$ that is optimal.

The principle

The polyhedron in standard form $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$ always has a vertex (if it's non-empty). If the problem

$$\min_x \quad c^T x \quad \text{such that } x \in \mathcal{P},$$

has a finite optimal cost, then there is an optimal vertex.

The principle

The polyhedron in standard form $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$ always has a vertex (if it's non-empty). If the problem

$$\min_x \quad c^T x \quad \text{such that } x \in \mathcal{P},$$

has a finite optimal cost, then there is an optimal vertex.

Principle: Move from one vertex of the polyhedron to an adjacent vertex of lower cost, until all adjacent vertices are of higher cost.

The principle

The polyhedron in standard form $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$ always has a vertex (if it's non-empty). If the problem

$$\min_{x} \quad c^T x \quad \text{such that } x \in \mathcal{P},$$

has a finite optimal cost, then there is an optimal vertex.

Principle: Move from one vertex of the polyhedron to an adjacent vertex of lower cost, until all adjacent vertices are of higher cost.

- How do you describe and find the vertices?
- How do you move from one vertex to an adjacent vertex?
- How do you select an adjacent vertex with a lower cost?
- Does the procedure converge? Towards an optimal solution?

Adjacent vertices

- Let $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$, and x^* and x^{**} be two basic feasible solutions of the polyhedron.

Adjacent vertices

- Let $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$, and x^* and x^{**} be two basic feasible solutions of the polyhedron.
- These two solutions are adjacent if there are $n-1$ linearly independent constraints active at both x^* and x^{**}.

Adjacent vertices

- Let $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$, and x^* and x^{**} be two basic feasible solutions of the polyhedron.
- These two solutions are adjacent if there are $n - 1$ linearly independent constraints active at both x^* and x^{**}.
- The m constraints $Ax = b$ are satisfied at x^* and x^{**}.

Adjacent vertices

- Let $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$, and x^* and x^{**} be two basic feasible solutions of the polyhedron.
- These two solutions are adjacent if there are $n-1$ linearly independent constraints active at both x^* and x^{**}.
- The m constraints $Ax = b$ are satisfied at x^* and x^{**}.

Adjacent vertices

- Let $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$, and x^* and x^{**} be two basic feasible solutions of the polyhedron.
- These two solutions are adjacent if there are $n-1$ linearly independent constraints active at both x^* and x^{**}.
- The m constraints $Ax = b$ are satisfied at x^* and x^{**}.
- There are $n-m$ constraints $x_i \geqslant 0$ tight in x^* and $n-m$ constraints $x_i \geqslant 0$ tight in x^{**}.

Adjacent vertices

- Let $\mathcal{P} = \{x \mid Ax = b, x \geqslant 0\}$, and x^* and x^{**} be two basic feasible solutions of the polyhedron.
- These two solutions are adjacent if there are $n-1$ linearly independent constraints active at both x^* and x^{**}.
- The m constraints $Ax = b$ are satisfied at x^* and x^{**}.
- There are $n-m$ constraints $x_i \geqslant 0$ tight in x^* and $n-m$ constraints $x_i \geqslant 0$ tight in x^{**}.
- The basic feasible solutions are adjacent if there are $n-m-1$ constraints $x_i \geqslant 0$ that are tight at both x^* and x^{**}.
- *This condition is satisfied if and only if the basic feasible x^* and x^{**} have $m-1$ basis variables in common.*

Example

The vertices of the polyhedron defined by

$$\begin{pmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & -3 & 1 \end{pmatrix} x = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad x \geqslant 0,$$

are (1, 1, 0, 0), (2, 0, 0, 1), (0, 8/5, 1/5, 0), (0, 0, 1, 4).

Example

The vertices of the polyhedron defined by

$$\begin{pmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & -3 & 1 \end{pmatrix} x = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad x \geqslant 0,$$

are (1, 1, 0, 0), (2, 0, 0, 1), (0, 8/5, 1/5, 0), (0, 0, 1, 4).

The vertex (1, 1, 0, 0) is adjacent to (2, 0, 0, 1) and to (0, 8/5, 1/5, 0) but is not adjacent to (0, 0, 1, 4).

The Simplex Method in Tabular Form

Let be the linear program

$$\begin{aligned} \min_x \quad & c^T x \\ \text{such that} \quad & Ax = b, \\ & x \geqslant 0. \end{aligned}$$

The problem is equivalent to that of minimizing the new variable z under the constraints

$$\begin{aligned} c^T x &= z, \\ Ax &= b, \\ x &\geqslant 0. \end{aligned}$$

- We're looking for the smallest value of z for which this polyhedron is not empty.
- In the following, we only write the equality constraints and always seek to minimize z.

The Canonical Tabular Form

The Simplex Method in Tabular Form - Example

Let be the linear program

$$
\begin{aligned}
\min_x \quad & x_2 - 5x_3 + 5x_4 & \\
& x_1 + x_2 - 11x_3 + 7x_4 & = 10, \\
& x_2 - 8x_3 + 4x_4 & = 4, \\
& x_1, x_2, x_3, x_4 & \geqslant 0,
\end{aligned}
$$

The Simplex Method in Tabular Form - Example

Let be the linear program

$$\begin{aligned} \min_x \quad & x_2 - 5x_3 + 5x_4 & \\ & x_1 + x_2 - 11x_3 + 7x_4 &= 10, \\ & x_2 - 8x_3 + 4x_4 &= 4, \\ & x_1, x_2, x_3, x_4 &\geqslant 0, \end{aligned}$$

written as the problem of the minimization of z under the constraints

$$\begin{aligned} x_2 - 5x_3 + 5x_4 &= z, \\ x_1 + x_2 - 11x_3 + 7x_4 &= 10, \\ x_1 - 8x_2 + 4x_4 &= 4, \\ x_1, x_2, x_3, x_4 &\geqslant 0. \end{aligned}$$

The Simplex Method in Tabular Form - Example

The Simplex Method in Tabular Form is given by

x_1	x_2	x_3	x_4	
0	1	-5	5	z
1	1	-11	7	10
0	1	-8	4	4

Example: Optimal distribution of resources between competing activities

- **Context**: Wood and nails are available to build tables and chairs.
- **Unit requirements**: you need 3 units of nails and 2 units of wood to build a chair, and 4 units of nails and 5 units of wood to build a table.
- **Limited resources**: We have 1700 units of nails and 1600 units of wood.
- **Unit profits**: A chair yields 2, a table 4.
- **Problem**: How many chairs and tables can you produce to maximize your profit?

The Simplex Method in Tabular Form - Example

The linear program is

$$
\begin{aligned}
\min_{x} \quad & -2x_1 - 4x_2 & \\
& 3x_1 + 4x_2 & \leqslant 1700, \\
& 2x_1 + 5x_2 & \leqslant 1600, \\
& x_1, x_2 & \geqslant 0.
\end{aligned}
$$

The Simplex Method in Tabular Form - Example

The linear program is

$$
\begin{aligned}
\min_{x} \quad & -2x_1 - 4x_2 \\
& 3x_1 + 4x_2 \leqslant 1700, \\
& 2x_1 + 5x_2 \leqslant 1600, \\
& x_1, x_2 \geqslant 0.
\end{aligned}
$$

The associated simplex table is given by

x_1	x_2	x_3	x_4	
-2	-4	0	0	z
3	4	1	0	1700
2	5	0	1	1600

Canonical form

Definition

The simplex table is in canonical form if

1. The matrix of constraints associated with the basic variables is the identity matrix.
2. basic variables do not appear in the objective function.

Canonical form

Definition

The simplex table is in canonical form if

1. The matrix of constraints associated with the basic variables is the identity matrix.
2. basic variables do not appear in the objective function.

This canonical form is obtained by elementary operations on the rows.

Canonical form

Definition

The simplex table is in canonical form if

1. The matrix of constraints associated with the basic variables is the identity matrix.
2. basic variables do not appear in the objective function.

This canonical form is obtained by elementary operations on the rows.

For example, the simplex table

x_1	x_2	x_3	x_4	
-2	-4	0	0	z
3	4	1	0	1700
2	5	0	1	1600

is in canonical form with basic variables $\{3, 4\}$.

Example

Let

$$
\begin{array}{ccccccccc}
 & & x_2 & - & 5x_3 & + & 5x_4 & = & z, \\
x_1 & + & x_2 & - & 11x_3 & + & 7x_4 & = & 10, \\
 & & x_2 & - & 8x_3 & + & 4x_4 & = & 4, \\
x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0.
\end{array}
$$

[2] Last lecture: we called such point a Basic Feasible Solution (BFS), associated to the basis $\{1, 2\}$.

Example

Let

$$\begin{array}{rcrcrcrcr} & & x_2 & - & 5x_3 & + & 5x_4 & = & z, \\ x_1 & + & x_2 & - & 11x_3 & + & 7x_4 & = & 10, \\ & & x_2 & - & 8x_3 & + & 4x_4 & = & 4, \\ x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0. \end{array}$$

The problem in canonical $\{1, 2\}$ form is given by

$$\begin{array}{rcrcrcrcr} & & & & 3x_3 & + & x_4 & = & z - 4, \\ x_1 & & & - & 3x_3 & + & 3x_4 & = & 6, \\ & & x_2 & - & 8x_3 & + & 4x_4 & = & 4, \\ x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0. \end{array}$$

This implies that the basic solution (6,4,0,0) is feasible ! [2]

[2] Last lecture: we called such point a Basic Feasible Solution (BFS), associated to the basis $\{1, 2\}$.

Example

The problem in canonical $\{1, 3\}$ form is given by

$$\begin{array}{rrrrrrrcl} & & 3/8x_2 & & & + & 5/2x_4 & = & z - 5/2, \\ x_1 & - & 3/8x_2 & & & + & 3/2x_4 & = & 9/2, \\ & - & 1/8x_2 & + & x_3 & - & 1/2x_4 & = & -1/2, \\ x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0. \end{array}$$

Example

The problem in canonical $\{1,3\}$ form is given by

$$
\begin{array}{ccccccccc}
 & & 3/8x_2 & & & + & 5/2x_4 & = & z - 5/2, \\
x_1 & - & 3/8x_2 & & & + & 3/2x_4 & = & 9/2, \\
 & - & 1/8x_2 & + & x_3 & - & 1/2x_4 & = & -1/2, \\
x_1 & , & x_2 & , & x_3 & , & x_4 & \geqslant & 0.
\end{array}
$$

The basic solution $\{1,3\}$ is not feasible.

Canonical form

Let x_B be the basic variables and x_N the non-basic variables.

$$\begin{aligned} c_B^T x_B + c_N^T x_N &= z, \\ A_B x_B + A_N x_N &= b, \\ x_B, x_N &\geqslant 0. \end{aligned}$$

Canonical form

Let x_B be the basic variables and x_N the non-basic variables.

$$\begin{array}{rcl} c_B^T x_B + c_N^T x_N & = & z, \\ A_B x_B + A_N x_N & = & b, \\ x_B, x_N & \geqslant & 0. \end{array}$$

If A_B is invertible, the problem can also be written as

$$\begin{array}{rcl} c_B^T x_B + c_N^T x_N & = & z, \\ x_B + A_B^{-1} A_N x_N & = & A_B^{-1} b, \\ x_B, x_N & \geqslant & 0. \end{array}$$

Canonical form

Let x_B be the basic variables and x_N the non-basic variables.

$$\begin{aligned} c_B^T x_B + c_N^T x_N &= z, \\ A_B x_B + A_N x_N &= b, \\ x_B, x_N &\geqslant 0. \end{aligned}$$

If A_B is invertible, the problem can also be written as

$$\begin{aligned} c_B^T x_B + c_N^T x_N &= z, \\ x_B + A_B^{-1} A_N x_N &= A_B^{-1} b, \\ x_B, x_N &\geqslant 0. \end{aligned}$$

We then have $x_B = A_B^{-1} b - A_B^{-1} A_N x_N$ and

$$\begin{aligned} c_B^T (A_B^{-1} b - A_B^{-1} A_N x_N) + c_N^T x_N &= z, \\ x_B + A_B^{-1} A_N x_N &= A_B^{-1} b, \\ x_B, x_N &\geqslant 0. \end{aligned}$$

Canonical form

Or equivalently

$$\begin{aligned}
(c_N^T - c_B^T A_B^{-1} A_N) x_N &= z - c_B^T A_B^{-1} b, \\
x_B + A_B^{-1} A_N x_N &= A_B^{-1} b, \\
x_B, x_N &\geqslant 0.
\end{aligned}$$

Canonical form

Or equivalently

$$
\begin{aligned}
(c_N^T - c_B^T A_B^{-1} A_N) x_N &= z - c_B^T A_B^{-1} b, \\
x_B + A_B^{-1} A_N x_N &= A_B^{-1} b, \\
x_B, x_N &\geqslant 0.
\end{aligned}
$$

- The basic solution is feasible if $A_B^{-1} b \geqslant 0$.
- It is optimal if the reduced costs $c_N^T - c_B^T A_B^{-1} A_N$ are positive (why? see further).

Canonical form

Or equivalently

$$\begin{array}{rcl}(c_N^T - c_B^T A_B^{-1} A_N) x_N & = & z - c_B^T A_B^{-1} b, \\ x_B + A_B^{-1} A_N x_N & = & A_B^{-1} b, \\ x_B, x_N & \geqslant & 0.\end{array}$$

- The basic solution is feasible if $A_B^{-1} b \geqslant 0$.
- It is optimal if the reduced costs $c_N^T - c_B^T A_B^{-1} A_N$ are positive (why? see further).

The principle of Simplex Method

The simplex algorithm starts from a vertex $(A_B^{-1} b \geqslant 0)$[a] and moves from vertex to vertex until an optimal vertex is reached $(c_N^T - c_B^T A_B^{-1} A_N \geqslant 0)$.

[a] A BFS then

Why the Canonical form ?

Let the simplex table be in canonical form with basis $\{1, 2\}$.

$$
\begin{array}{cccc|c}
0 & 0 & c_3 & c_4 & z-4 \\
\hline
1 & 0 & -3 & 3 & 6 \\
0 & 1 & -8 & 4 & 4
\end{array}
$$

The quantities shown in the first row of the table are the reduced costs of the corresponding variables. Depending on the values of the reduced costs, there are three possible cases:

1. The vertex (6,4,0,0) is optimal.
2. The vertex (6,4,0,0) is the end of a half-right totally contained in the polyhedron and along which the cost is decreasing. The cost is unbounded.
3. The vertex is adjacent to a lower-cost vertex.

These three cases can be read directly from the table !

Case 1 - Optimal Vertex

$$\begin{array}{cccc|c} 0 & 0 & 3 & 1 & z-4 \\ \hline 1 & 0 & -3 & 3 & 6 \\ 0 & 1 & -8 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1,2\}$ basis.

Case 1 - Optimal Vertex

$$\begin{array}{cccc|c} 0 & 0 & 3 & 1 & z-4 \\ \hline 1 & 0 & -3 & 3 & 6 \\ 0 & 1 & -8 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

- From the first row: we have $3x_3 + x_4 = z - 4$.
- In (6,4,0,0), z is equal to 4.
- Since $x_3, x_4 \geqslant 0$, we have $3x_3 + x_4 \geqslant 0$ and the cost of another feasible solution can only be greater than or equal to 4 !

Case 1 - Optimal Vertex

$$
\begin{array}{cccc|c}
0 & 0 & 3 & 1 & z-4 \\
\hline
1 & 0 & -3 & 3 & 6 \\
0 & 1 & -8 & 4 & 4
\end{array}
$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

- From the first row: we have $3x_3 + x_4 = z - 4$.
- In (6,4,0,0), z is equal to 4.
- Since $x_3, x_4 \geqslant 0$, we have $3x_3 + x_4 \geqslant 0$ and the cost of another feasible solution can only be greater than or equal to 4 !
  Indeed: $z = 4 + \underbrace{3x_3 + x_4}_{\geqslant 0} \geqslant 4$ and the vertex (6,4,0,0) is optimal with cost 4.

Case 1 - Optimal Vertex

0	0	3	1	$z-4$
1	0	-3	3	6
0	1	-8	4	4

(6,4,0,0) is the BFS associated with the $\{1,2\}$ basis.

- From the first row: we have $3x_3 + x_4 = z - 4$.
- In (6,4,0,0), z is equal to 4.
- Since $x_3, x_4 \geqslant 0$, we have $3x_3 + x_4 \geqslant 0$ and the cost of another feasible solution can only be greater than or equal to 4 !
  Indeed: $z = 4 + \underbrace{3x_3 + x_4}_{\geqslant 0} \geqslant 4$ and the vertex (6,4,0,0) is optimal with cost 4.
- At vertex (6,4,0,0), the reduced cost of x_3 is equal to 3, and of x_4 to 1.

If at a vertex all reduced costs are positive or zero, then the vertex is optimal.

Case 2 - The cost is unbounded.

$$\begin{array}{cccc|c} 0 & 0 & -3 & 1 & z-4 \\ \hline 1 & 0 & -3 & 3 & 6 \\ 0 & 1 & -8 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

Case 2 - The cost is unbounded.

$$\begin{array}{cccc|c} 0 & 0 & -3 & 1 & z-4 \\ \hline 1 & 0 & -3 & 3 & 6 \\ 0 & 1 & -8 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1,2\}$ basis.

- The reduced cost of x_3 is negative.
- If x_3 increases and x_4 remains unchanged, the cost $(z = 4 - 3x_3 + x_4)$ decreases, whatever the evolution of x_1 and x_2.

Case 2 - The cost is unbounded.

$$\begin{array}{cccc|c} 0 & 0 & -3 & 1 & z-4 \\ \hline 1 & 0 & -3 & 3 & 6 \\ 0 & 1 & -8 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

- The reduced cost of x_3 is negative.
- If x_3 increases and x_4 remains unchanged, the cost $(z = 4 - 3x_3 + x_4)$ decreases, whatever the evolution of x_1 and x_2.
- Keep $x_4 = 0$ and set $x_3 = \lambda$. We get $z = 4 - 3\lambda$. As λ increases, z decreases.
- What about the constraints ?
  - The two constraints become $x_1 = 6 + 3\lambda$ and $x_2 = 4 + 8\lambda$.

Case 2 - The cost is unbounded.

$$\begin{array}{cccc|c} 0 & 0 & -3 & 1 & z-4 \\ \hline 1 & 0 & -3 & 3 & 6 \\ 0 & 1 & -8 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

- The reduced cost of x_3 is negative.
- If x_3 increases and x_4 remains unchanged, the cost $(z = 4 - 3x_3 + x_4)$ decreases, whatever the evolution of x_1 and x_2.
- Keep $x_4 = 0$ and set $x_3 = \lambda$. We get $z = 4 - 3\lambda$. As λ increases, z decreases.
- What about the constraints ?
  - The two constraints become $x_1 = 6 + 3\lambda$ and $x_2 = 4 + 8\lambda$.
  - As λ increases, we always satisfy $x_1, x_2 \geqslant 0$.

Case 2 - The cost is unbounded.

$$\begin{array}{cccc|c} 0 & 0 & -3 & 1 & z-4 \\ \hline 1 & 0 & -3 & 3 & 6 \\ 0 & 1 & -8 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

- The reduced cost of x_3 is negative.
- If x_3 increases and x_4 remains unchanged, the cost $(z = 4 - 3x_3 + x_4)$ decreases, whatever the evolution of x_1 and x_2.
- Keep $x_4 = 0$ and set $x_3 = \lambda$. We get $z = 4 - 3\lambda$. As λ increases, z decreases.
- What about the constraints ?
  - The two constraints become $x_1 = 6 + 3\lambda$ and $x_2 = 4 + 8\lambda$.
  - As λ increases, we always satisfy $x_1, x_2 \geqslant 0$.
  - The half-right $(6 + 3\lambda, 4 + 8\lambda, \lambda, 0)$ for $\lambda \geqslant 0$ is totally contained in the polyhedron and the cost is decreasing along the half-right.

The cost is not bounded - If at a vertex, one of the reduced costs is strictly negative, and the entries in the corresponding column in the simplex table are negative, then the cost is not bounded $(f^* = -\infty)$.

Case 3 - Lower-cost adjacent vertex.

$$
\begin{array}{cccc|c}
0 & 0 & -3 & -1 & z-4 \\
\hline
1 & 0 & 3 & 3 & 6 \\
0 & 1 & 6 & 4 & 4
\end{array}
$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

Case 3 - Lower-cost adjacent vertex.

$$\begin{array}{cccc|c} 0 & 0 & -3 & -1 & z-4 \\ \hline 1 & 0 & 3 & 3 & 6 \\ 0 & 1 & 6 & 4 & 4 \end{array}$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

- The reduced costs associated with x_3 and x_4 are negative. If x_3 or x_4 increases, the cost decreases, whatever the evolution of the variables x_1 and x_2.

Case 3 - Lower-cost adjacent vertex.

$$
\begin{array}{cccc|c}
0 & 0 & -3 & -1 & z-4 \\
\hline
1 & 0 & 3 & 3 & 6 \\
0 & 1 & 6 & 4 & 4
\end{array}
$$

(6,4,0,0) is the BFS associated with the $\{1, 2\}$ basis.

- The reduced costs associated with x_3 and x_4 are negative. If x_3 or x_4 increases, the cost decreases, whatever the evolution of the variables x_1 and x_2.
- We choose to introduce x_3 into the basis, and keep x_4 outside the basis: $x_3 = \lambda \geqslant 0$ and $x_4 = 0$.
- *In order to find a new vertex, we need to determine the variable that leaves the basis.*
- **Constraints**: become $x_1 + 3\lambda = 6$ and $x_2 + 6\lambda = 4$.
  - As λ increases, the constraint $x_2 + 6\lambda = 4$ is the first to become critical. It's x_2 that leaves the basis.
  - We obtain $\lambda = 2/3$ and move on to the vertex $(4, 0, 2/3, 0)$. The new basis is $\{1, 3\}$.
- The cost decreases by 2.

The Simplex algorithm

Simplex algorithm

Simplex algorithm

Given a basic feasible solution

1. Write the simplex table in canonical form.
2. If all reduced costs are positive or zero, stop (x is optimal). If not, choose an non-basic variable x_k with a negative reduced cost.
3. Let a be the column of the simplex table associated with the variable x_k and d the last column of the table:
    - 3.1 If $a \leqslant 0$, stop (optimal cost is not bounded).
    - 3.2 Otherwise, there is at least one index i for which $a_i > 0$
        - 3.2.1 Calculate the quotients $\frac{d_i}{a_i}$ for indices i for which $a_i > 0$.
        - 3.2.2 find the index l of the smallest quotient: l is such that $a_l > 0$ and

$$\frac{d_l}{a_l} \leqslant \frac{d_i}{a_i} \text{ for all } i \text{ such that } a_i > 0.$$

4. The variable x_k enters the base and x_l leaves it. Update the base. Return to 1.

Simplex algorithm

Simplex algorithm

Given a basic feasible solution

1. Write the simplex table in canonical form.
2. If all reduced costs are positive or zero, stop (x is optimal). If not, choose an non-basic variable x_k with a negative reduced cost.
3. Let a be the column of the simplex table associated with the variable x_k and d the last column of the table:
   3.1 If $a \leqslant 0$, stop (optimal cost is not bounded).
   3.2 Otherwise, there is at least one index i for which $a_i > 0$
      3.2.1 Calculate the quotients $\frac{d_i}{a_i}$ for indices i for which $a_i > 0$.
      3.2.2 find the index l of the smallest quotient: l is such that $a_l > 0$ and

$$\frac{d_l}{a_l} \leqslant \frac{d_i}{a_i} \text{ for all } i \text{ such that } a_i > 0.$$

4. The variable x_k enters the base and x_l leaves it. Update the base. Return to 1.

An iteration is also called a pivot.

Algorithm convergence

- At step 2: which variable to choose from among the non-basic variables for negative reduced cost?

Algorithm convergence

- At step 2: which variable to choose from among the non-basic variables for negative reduced cost?
- At each stage we are free to choose the variable we prefer:
  1. the minimum reduced cost variable,
  2. the variable that most reduces the cost function,
  3. the negative reduced cost variable with the smallest index, etc.

Algorithm convergence

- At step 2: which variable to choose from among the non-basic variables for negative reduced cost?
- At each stage we are free to choose the variable we prefer:
  1. the minimum reduced cost variable,
  2. the variable that most reduces the cost function,
  3. the negative reduced cost variable with the smallest index, etc.
- *If the* ***cost strictly decreases*** *with each pivot, then the algorithm converges.*

Algorithm convergence

- At step 2: which variable to choose from among the non-basic variables for negative reduced cost?
- At each stage we are free to choose the variable we prefer:
  1. the minimum reduced cost variable,
  2. the variable that most reduces the cost function,
  3. the negative reduced cost variable with the smallest index, etc.
- *If the* ***cost strictly decreases*** *with each pivot, then the algorithm converges.*
- Indeed:
  - at each iteration, we reach a new vertex whose cost is strictly lower than the cost of the preceding vertices.
    So we can't go over the same vertex twice.
  - On the other hand, a polyhedron has only a finite number of vertices, so the algorithm must stop.

Example

$$
\begin{aligned}
\min_x \ & -2x_1 - 4x_2 \\
\text{such that} \quad & 3x_1 + 4x_2 \leqslant 1700, \\
& 2x_1 + 5x_2 \leqslant 1600, \\
& x_1, x_2 \geqslant 0.
\end{aligned}
$$

The associated simplex table is given by

x_1	x_2	x_3	x_4	
-2	-4	0	0	z
3	4	1	0	1700
2	5	0	1	1600

Example

Iteration 1.

- The solution (0, 0, 1700, 1600) is the BFS associated with the basic variables x_3 and x_4.
- The reduced costs associated with the variables x_1 and x_2 are negative.
- We choose to enter x_2 in the base.
- We can't increase x_2 without limit, since we have to satisfy the constraints:
  1. The first imposes $4x_2 + x_3 = 1700$.
  2. and the second $5x_2 + x_4 = 1600$.

  The second constraint is the most restrictive. It's x_4 that leaves the base.
- After elementary transformations on the rows, we obtain the canonical table

x_1	x_2	x_3	x_4	
$-2/5$	0	0	$4/5$	$z + 1280$
$7/5$	0	1	$-4/5$	420
$2/5$	1	0	$1/5$	320

Example

Iteration 2.

- The solution (0, 320, 420, 0) is the new BFS associated with the basic variables x_2 and x_3. The non-basic variables are x_1 and x_4.
- The cost increases if x_4 increases, decreases if x_1 increases. We enter x_1 in the base.
- Which of the variables x_2 and x_3 leaves the base? The constraints are
  - $7/5x_1 + x_3 = 420$, and
  - $2/5x_1 + x_2 = 320$.

  The most restrictive of constraints is the first !
- we obtain, after transformation,

x_1	x_2	x_3	x_4	
0	0	$2/7$	$4/7$	$z + 1400$
1	0	$5/7$	$-4/7$	300
0	1	$-2/7$	$3/7$	200

The non-basic variables are x_3 and x_4.

The associated reduced costs are positive. The solution is optimal !

Degenerate Case and Remedy

Degenerate case. Example

Consider the linear program

$$
\begin{aligned}
\min_{x \geqslant 0} \quad & -3x_1 + x_2 \\
\text{such that} \quad & 2x_1 - x_2 \leqslant 4, \\
& x_1 - 2x_2 \leqslant 2, \\
& x_1 + x_2 \leqslant 5.
\end{aligned}
$$

Or minimize z with

$$
\begin{array}{ccccccccccccc}
-3x_1 & + & x_2 & & & & & & & = & z, \\
2x_1 & - & x_2 & + & x_3 & & & & & = & 4, \\
x_1 & - & 2x_2 & & & + & x_4 & & & = & 2, \\
x_1 & + & x_2 & & & & & + & x_5 & = & 5, \\
x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geqslant & 0.
\end{array}
$$

Degenerate case. Example - Iteration 1.

x_1	x_2	x_3	x_4	x_5	
-3	1	0	0	0	z
2	-1	1	0	0	4
1	-2	0	1	0	2
1	1	0	0	1	5

- **Context**:(0,0,4,2,5) is a BFS associated with the $\{3, 4, 5\}$ basis. The reduced cost associated with x_1 is negative, so we enter it in the base.
- For $x_1 = \lambda \geqslant 0$ and $x_2 = 0$, the constraints are as follows:
  1. $x_3 = 4 - 2\lambda \geqslant 0$, and
  2. $x_4 = 2 - \lambda \geqslant 0$, and
  3. $x_5 = 5 - \lambda \geqslant 0$.
- The first two constraints are activated in λ =2: the variables x_3 and x_4 are **both** candidates for leaving the base.
- Whichever variable we choose, the other variable will be zero at the next iteration, and we'll find ourselves in a **degenerate** vertex. We choose to output x_4.

Degenerate case. Example - Iteration 2.

x_1	x_2	x_3	x_4	x_5	
0	-5	0	3	0	$z+6$
0	3	1	-2	0	0
1	-2	0	1	0	2
0	3	0	-1	1	3

- The x_2 variable is the only non-basic variable with a negative reduced cost.
- The variable x_2 enters the base.
- The only possibility is to move x_3 out of the base, and we have the base $\{1, 2, 5\}$.

Degenerate case. Example - Iteration 3.

x_1	x_2	x_3	x_4	x_5	
0	0	$5/3$	$-1/3$	0	$z+6$
0	1	$1/3$	$-2/3$	0	0
1	0	$2/3$	$-1/3$	0	2
0	0	-1	1	1	3

- The base has changed but the BFS has not.
- In particular, the cost has not decreased.

Degenerate case. Example - Iteration 4.

The next iteration is more standard: x_4 enters the base and x_5 exits.

x_1	x_2	x_3	x_4	x_5	
0	0	$4/3$	0	$1/3$	$z+7$
0	1	$-1/3$	0	$2/3$	2
1	0	$1/3$	0	$1/3$	3
0	0	-1	1	1	3

- Solution (3, 2, 0, 3, 0) is optimal. The optimal cost is -7.
- **Solution in CVXPY**: see colab file - Section 1

Cycling

- At iteration 3 of the example, the cost has not decreased as a result of the pivot.
- This situation is **only** possible when you're at a **degenerate** vertex.

Cycling

- At iteration 3 of the example, the cost has not decreased as a result of the pivot.
- This situation is **only** possible when you're at a **degenerate** vertex.
- When the cost does not decrease after one or more successive pivots, *it is not excluded to return to a previously visited base* and cycling by moving onto degenerate vertices of equal cost.

Cycling

- At iteration 3 of the example, the cost has not decreased as a result of the pivot.
- This situation is **only** possible when you're at a **degenerate** vertex.
- When the cost does not decrease after one or more successive pivots, *it is not excluded to return to a previously visited base* and cycling by moving onto degenerate vertices of equal cost.

Cycling

- At iteration 3 of the example, the cost has not decreased as a result of the pivot.
- This situation is **only** possible when you're at a **degenerate** vertex.
- When the cost does not decrease after one or more successive pivots, *it is not excluded to return to a previously visited base* and cycling by moving onto degenerate vertices of equal cost.
- **Consequence**: the argument used to establish the convergence of the algorithm in the case of non-degenerate vertices was based on the decrease, at each iteration, of the cost function. *This argument is therefore invalid when the algorithm encounters a degenerate vertex.*

Cycling

- At iteration 3 of the example, the cost has not decreased as a result of the pivot.
- This situation is **only** possible when you're at a **degenerate** vertex.
- When the cost does not decrease after one or more successive pivots, *it is not excluded to return to a previously visited base* and cycling by moving onto degenerate vertices of equal cost.
- **Consequence**: the argument used to establish the convergence of the algorithm in the case of non-degenerate vertices was based on the decrease, at each iteration, of the cost function. *This argument is therefore invalid when the algorithm encounters a degenerate vertex.*

General Remedy

When we are dealing with degenerate vertices, the convergence of the algorithm is only guaranteed for particular pivoting strategies.

One Remedy - the Bland's rule

- Among the candidate variables *to enter the base*, select the one with the smallest index.
- Among the candidate variables *to leave the base*, select the one with the smallest index.

Reference: Robert G. Bland, *New finite pivoting rules for the simplex method*, Mathematics of Operations Research 2, pp. 103-107, 1977.

Initialization

General Comment

- The simplex algorithm requires starting from a *vertex* of the polyhedron, or equivalently a BFS.

General Comment

- The simplex algorithm requires starting from a *vertex* of the polyhedron, or equivalently a BFS.
- An initial vertex is not always available.

General Comment

- The simplex algorithm requires starting from a *vertex* of the polyhedron, or equivalently a BFS.
- An initial vertex is not always available.
- The search for a vertex of a polyhedron can be carried out during an initialization phase (or Phase I)
- Phase I consists in solving an **auxiliary** linear program, for which we **always** have an initial vertex.

The Auxiliary Linear Program

- Consider the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b, x \geqslant 0\}$ for which we're looking for a vertex.
- Without loss of generality, we can assume that $b \geqslant 0$ (otherwise we can multiply the equalities corresponding to $b_i < 0$ by -1).

The Auxiliary Linear Program

- Consider the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b, x \geqslant 0\}$ for which we're looking for a vertex.
- Without loss of generality, we can assume that $b \geqslant 0$ (otherwise we can multiply the equalities corresponding to $b_i < 0$ by -1).
- We introduce artificial variables y_i $1 \leqslant i \leqslant m$ and construct the auxiliary problem $(m + n$ variables):

$$
\begin{aligned}
\min_{x,y} \quad & \sum_i y_i \\
& Ax + y = b, \\
& x \geqslant 0, y \geqslant 0.
\end{aligned}
$$

The Auxiliary Linear Program

- Consider the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b, x \geqslant 0\}$ for which we're looking for a vertex.
- Without loss of generality, we can assume that $b \geqslant 0$ (otherwise we can multiply the equalities corresponding to $b_i < 0$ by -1).
- We introduce artificial variables y_i $1 \leqslant i \leqslant m$ and construct the auxiliary problem $(m + n$ variables):

$$\begin{aligned} \min_{x,y} \quad & \sum_i y_i \\ & Ax + y = b, \\ & x \geqslant 0, y \geqslant 0. \end{aligned}$$

- **Key observation**: solution $x = 0$ and $y = b$ is feasible and tightens $m + n$ lin. ind. constraints.
  It is therefore a vertex and we can start the simplex algorithm !!

The Auxiliary Linear Program

Two scenarios may occur:

The Auxiliary Linear Program

Two scenarios may occur:

1. If the optimal cost of the auxiliary problem is greater than 0, there is no feasible solution with $y = 0$,
   $\rightarrow$ the initial polyhedron $\mathcal{P}$ is empty.

The Auxiliary Linear Program

Two scenarios may occur:

1. If the optimal cost of the auxiliary problem is greater than 0, there is no feasible solution with $y = 0$,
   $\rightarrow$ the initial polyhedron $\mathcal{P}$ is empty.
2. If the optimal cost of the auxiliary problem is equal to 0 ($\iff y^* = 0$), we consider an optimal BFS $(x^*, 0)$.
   $\rightarrow$ the solution x^* is a vertex of the initial polyhedron $\mathcal{P}$. Why?

Example

To find a vertex of a polyhedron defined by

$$
\begin{array}{ccccc}
7x_1 & + & 6x_2 & = & 5, \\
-2x_1 & - & 4x_2 & \geqslant & -2, \\
x_1 & + & 5x_2 & \geqslant & 6, \\
x_1 & , & x_2 & \geqslant & 0,
\end{array}
$$

Example

To find a vertex of a polyhedron defined by

$$\begin{array}{rcrcr} 7x_1 & + & 6x_2 & = & 5, \\ -2x_1 & - & 4x_2 & \geqslant & -2, \\ x_1 & + & 5x_2 & \geqslant & 6, \\ x_1 & , & x_2 & \geqslant & 0, \end{array}$$

we first introduce slack variables

$$\begin{array}{rcrcrcrcr} 7x_1 & + & 6x_2 & & & & & = & 5, \\ -2x_1 & - & 4x_2 & - & s_1 & & & = & -2, \\ x_1 & + & 5x_2 & & & - & s_2 & = & 6, \\ x_1 & , & x_2 & , & s_1 & , & s_2 & \geqslant & 0, \end{array}$$

Example

and write the constraints with a positive b vector

$$\begin{array}{ccccccccc}
7x_1 & + & 6x_2 & & & & & = & 5, \\
2x_1 & + & 4x_2 & + & s_1 & & & = & 2, \\
x_1 & + & 5x_2 & & & - & s_2 & = & 6, \\
x_1 & , & x_2 & , & s_1 & , & s_2 & \geqslant & 0.
\end{array}$$

Example

and write the constraints with a positive b vector

$$\begin{array}{ccccccccc} 7x_1 & + & 6x_2 & & & & & = & 5, \\ 2x_1 & + & 4x_2 & + & s_1 & & & = & 2, \\ x_1 & + & 5x_2 & & & - & s_2 & = & 6, \\ x_1 & , & x_2 & , & s_1 & , & s_2 & \geqslant & 0. \end{array}$$

Finally, we introduce the artificial variables

$$\begin{array}{ccccccccccccccc} 7x_1 & + & 6x_2 & & & & & + & y_1 & & & & & = & 5, \\ 2x_1 & + & 4x_2 & + & s_1 & & & & & + & y_2 & & & = & 2, \\ x_1 & + & 5x_2 & & & - & s_2 & & & & & + & y_3 & = & 6, \\ x_1 & , & x_2 & , & s_1 & , & s_2 & , & y_1 & , & y_2 & , & y_3 & \geqslant & 0. \end{array}$$

- *Always an initial vertex*: (0, 0, 0, 0, 5, 2, 6) is a vertex of this polyhedron.
- We can start the simplex algorithm with the aim of minimizing $y_1 + y_2 + y_3$.

Example

Finally, the optimal solution to the problem

$$\min_{x,y} y_1 + y_2 + y_3$$

such that

$$\begin{array}{ccccccccccccccccc}
7x_1 & + & 6x_2 & & & & & + & y_1 & & & & & = & 5, \\
2x_1 & + & 4x_2 & + & s_1 & & & & & + & y_2 & & & = & 2, \\
x_1 & + & 5x_2 & & & - & s_2 & & & & & + & y_3 & = & 6, \\
x_1 & , & x_2 & , & s_1 & , & s_2 & , & y_1 & , & y_2 & , & y_3 & \geqslant & 0,
\end{array}$$

is $(0.5, 0.25, 0, 0, 0, 0, 4.25)$.

Conclusions ?

Example

Finally, the optimal solution to the problem

$$\min_{x,y} y_1 + y_2 + y_3$$

such that

$$\begin{array}{ccccccccccccccc}
7x_1 & + & 6x_2 & & & & & + & y_1 & & & & & = & 5, \\
2x_1 & + & 4x_2 & + & s_1 & & & & & + & y_2 & & & = & 2, \\
x_1 & + & 5x_2 & & & - & s_2 & & & & & + & y_3 & = & 6, \\
x_1 & , & x_2 & , & s_1 & , & s_2 & , & y_1 & , & y_2 & , & y_3 & \geqslant & 0,
\end{array}$$

is $(0.5, 0.25, 0, 0, 0, 0, 4.25)$.

Conclusions ?

- the original problem is infeasible/impossible (because $y^* \neq 0$).
- This avoids having to test all the bases (as a reminder, there's $C_n^m = \frac{n!}{m!(n-m)!}$ which grows exponentially) before you know it.
- *Solution in CVXPY*: ▸ see colab file - Section 2

Very brief comment on the Complexity

Complexity

- **The worst case**: We know how to construct problems of n variables for which the simplex algorithm generates $2^n - 1$ iterations. Such problems can be constructed for most of the pivoting rules used in practice.

[3] see Part 2 of this course if we are lucky

Complexity

- **The worst case**: We know how to construct problems of n variables for which the simplex algorithm generates $2^n - 1$ iterations. Such problems can be constructed for most of the pivoting rules used in practice.
- **The average case**: And yet, in practice, the simplex algorithm is commonly used to solve problems with thousands of variables. It turns out that the number of operations performed on average is polynomial in n and m

[3] see Part 2 of this course if we are lucky

Complexity

- **The worst case**: We know how to construct problems of n variables for which the simplex algorithm generates $2^n - 1$ iterations. Such problems can be constructed for most of the pivoting rules used in practice.
- **The average case**: And yet, in practice, the simplex algorithm is commonly used to solve problems with thousands of variables. It turns out that the number of operations performed on average is polynomial in n and m
- **Polynomial algorithm?**:
  - There are algorithms that always take polynomial time. These are known as Interior Point Methods[3].
  - However, they don't always work better than the simplex algorithm (depending on size, type of problem, etc.).

[3] see Part 2 of this course if we are lucky

Conclusions

Summary

We have seen

- **The principle** of the *Simplex Algorithm* : Move from one vertex of the polyhedron to an adjacent vertex of lower cost, until all adjacent vertices are of higher cost.
- The **Canonical Tabular** form: start with the standard form + introduction of z *AND*
  1. The matrix of constraints associated with the basis variables is the identity matrix.
  2. basis variables do not appear in the objective function.

  Why this form ? The *three* scenarios (optimal, unbounded and lower-cost adjacent vertex) *can be read directly* from the table !
- The Simplex **Algorithm** and convergence.
- The **Degenerate** case, the potential *cycling* and one *Remedy* (the Bland's rule).
- The **Initialization**: solve an *auxiliary* linear program to find a vertex of the original polyhedron $\mathcal{P}$.

Preparations for the next lecture

- Review the lecture :); many important results and notions have been introduced.
- Compute by hands the successive simplex tables (in canonical form !) for the examples in this course at:
  1. Slide 32 - use the Bland's rule
  2. Slide 43 - the auxiliary Problem, how many iterations ?

Goodbye, So Soon

THANKS FOR THE ATTENTION

- v.leplat@innopolis.ru
- sites.google.com/view/valentinleplat/