Discrete Optimization and Integer Programming.

Linear Programming

“Nothing takes place in the world whose meaning is not that of some maximum or minimum.” (L. Euler)

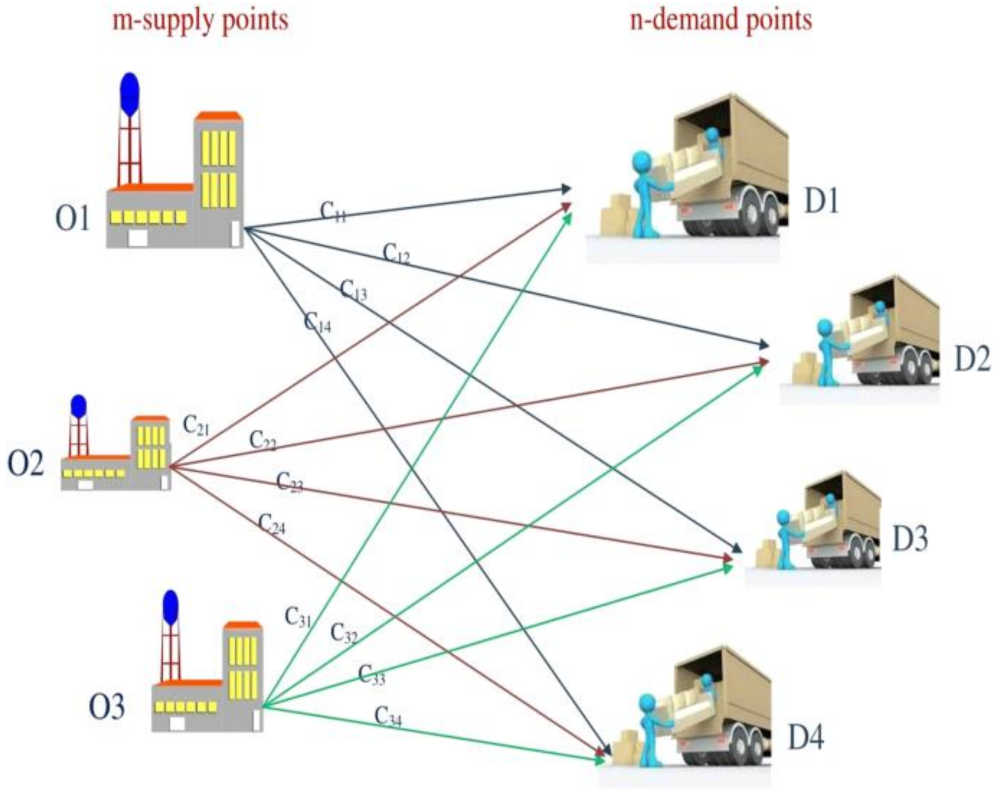
Contents

I. <u>Linear Programming</u>

- ***Transportation Problem***

The *<u>Transportation Problem</u>* is the classical linear optimization problem which can be formulated as follows:

Suppose we have m sources and n sinks for a commodity. For each source $i = 1, \dots, m$ and for each sink $j = 1, \dots, m$ the following constants are given: production supply a_i at the source i, demand b_j at the sink j, the unit cost c_{ij} of shipment from the source i to the sink j. It is assumed that the total production is equal to the total consumption: $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$. It is required to make a transportation plan $\{x_{ij}\}$ that allows the products of all manufacturers (sources) to be completely exported, fully meeting the needs of all consumers (sinks) and giving a minimum of total transportation cost.



$$\sum_{j=1}^{n} x_{ij} = a_i, \qquad i = 1, \dots, m,$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \qquad j = 1, \dots, n,$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min.$$

I. Linear Programming

- ***Transportation Problem***

Suppose some company rents cars. It detects imbalance in distribution of cars in 6 cities. In the cities 1, 2, 3 there is an excess of cars (26, 43, 31, respectively) and in the cities 4, 5, 6, there is a lack of cars (32, 5, 6, respectively). The cost of driving a car from the city i to the city j is proportional to the distance between these cities which is equal to c_{ij}. We need to form a plan of most economical relocation of cars. It leads to the following LP problem.

$$
\begin{gathered}
\sum_{i,j} c_{ij} x_{ij} \longrightarrow \min \\
x_{14} + x_{15} + x_{16} = 26, \\
x_{24} + x_{25} + x_{26} = 43, \\
x_{34} + x_{35} + x_{36} = 31, \\
x_{14} + x_{24} + x_{34} = 32, \\
x_{15} + x_{25} + x_{35} = 5, \\
x_{16} + x_{26} + x_{36} = 6, \\
x_{14}, x_{15}, x_{16}, \dots, x_{34}, x_{35}, x_{36} \geq 0.
\end{gathered}
$$

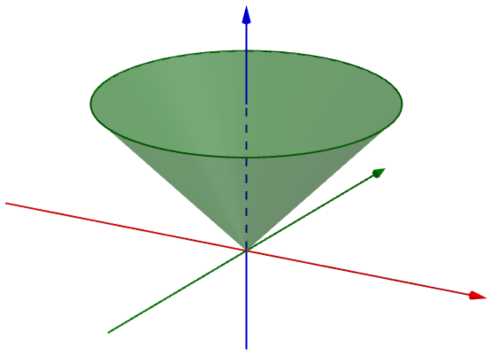
I. Linear Programming

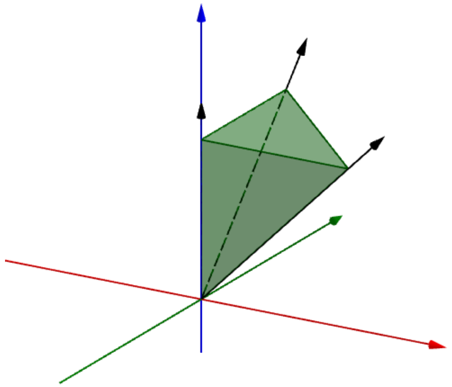
• *Basic Definitions*

There are the following equivalent formulations of *Linear Optimization Problem*:

$$\max\{c \cdot x \mid Ax \le b\}, \qquad \max\{c \cdot x \mid x \ge 0, Ax \le b\}, \qquad \max\{c \cdot x \mid x \ge 0, Ax = b\}, \ldots$$

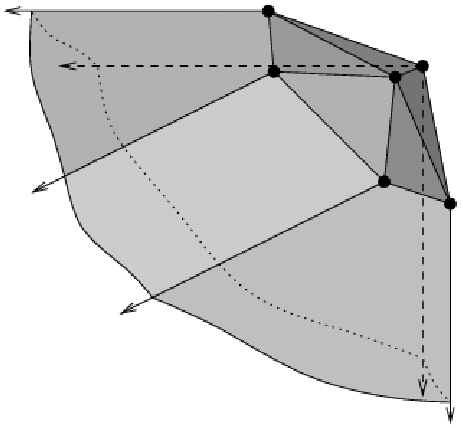
The *feasibility domain* for LP problem determines a polyhedron which we will denote by P.

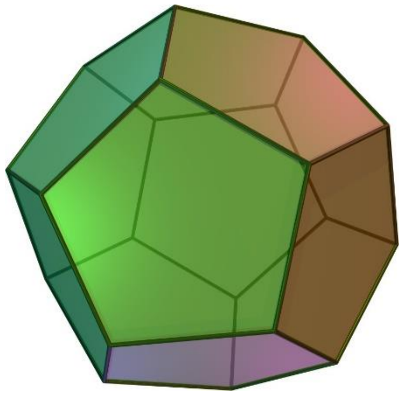




Definition 1. A non-empty set C of vectors in $\mathbb{R}^n$ is called *convex cone* if $\lambda x + \mu y \in C$ for all $x, y \in C$ and $\lambda, \mu \ge 0$.

Definition 2. A cone C is called *polyhedral* if $C = \{x \mid Ax \le 0\}$ for some matrix A.





Definition 3. A set P of vectors in $\mathbb{R}^n$ is called *polyhedron* if $P = \{x \mid Ax \le b\}$ for some matrix A and vector b.

Definition 4. A set of vectors is called *convex polytope* if it is a convex hull of finite number of vectors.

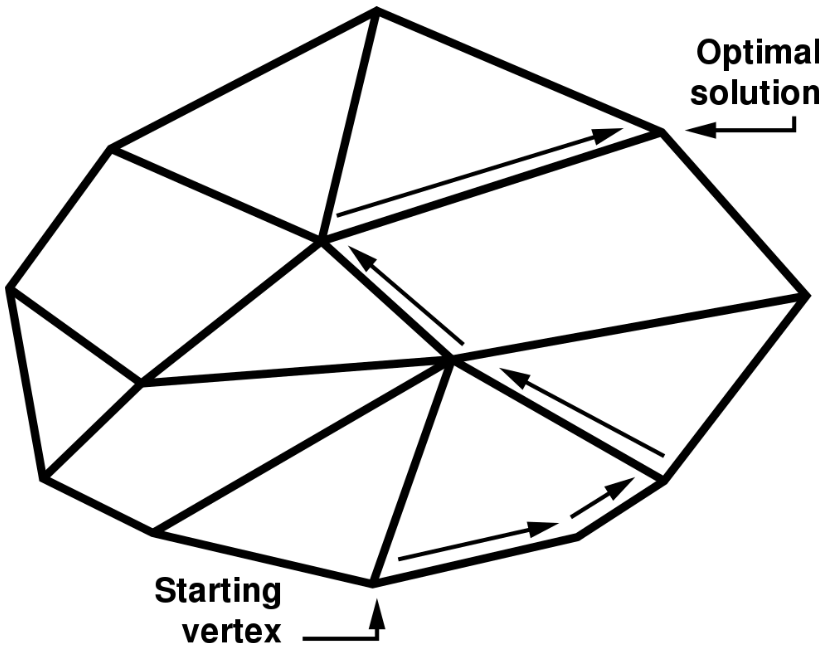
II. Dantzig’s Simplex Algorithm

• *Geometric Idea*

Definition 5. A *basic feasible solution* of an LP problem is a vertex of the corresponding polyhedron P. Each feasible solution corresponds to the choice of subset of the set of variables. This subset is called a *basis* and the corresponding variables are called *basic variables*.

The *Dantzig’s Simplex Algorithm* is a famous algorithm for linear programming. The geometric idea of this algorithm is the following

1. Find any *basic feasible solution* of the LP problem;
2. Look at the neighboring vertices;
3. Two cases can occur:
   - If there is a better neighboring vertex (it increases objective function and the basic variables remain feasible), move to the best one and go to the step 2.
   - If there is no better vertex, then the current vertex is the solution or there is no solution at all.



II. Dantzig's Simplex Algorithm

• *Algorithm's Description*

Consider an LP problem in the form

$$\max\{c \cdot x \mid x \geq 0, Ax = b\} \text{ such that } x \in \mathbb{R}^n, A \in \mathrm{Mat}^{m\times n}(\mathbb{R}), \mathrm{rank}(A) = \mathrm{m}.$$

Then the algorithm can be described as follows.

I. Choose a subset $J \in \{1, \dots, m\} = N, |J| = n$ of indices of variables such that the $n \times n$ sub-matrix A_J is invertible. Then we have $x_J = A_J^{-1}b - A_J^{-1}A_{N\setminus \mathrm{J}}x_{N\setminus \mathrm{J}}$, so the solution $x_J = A_J^{-1}b, x_{N\setminus \mathrm{J}} = 0$ is a *basic solution*. If $A_J^{-1}b \geq 0$ then it is a *feasible basic solution* and we take this as an initial solution for the algorithm.

II. The objective function in the chosen basis can be expressed as follows

$$c \cdot x = c_J x_J + c_{N\setminus \mathrm{J}}x_{N\setminus \mathrm{J}} = c_J A_J^{-1}b + h_{N\setminus \mathrm{J}}x_{N\setminus \mathrm{J}}, \qquad h_{N\setminus \mathrm{J}} = c_{N\setminus \mathrm{J}} - c_J A_J^{-1}A_{N\setminus \mathrm{J}}.$$

- If the reduced cost vector $h_{N\setminus \mathrm{J}} < 0$ then the solution $x_J = A_J^{-1}b, x_{N\setminus \mathrm{J}} = 0$ is optimal.
- If there are positive components of the vector $h_{N\setminus \mathrm{J}}$ then we can improve solution by varying non-basic variables $x_{N\setminus \mathrm{J}}$. We do this in the step III.

II. Dantzig’s Simplex Algorithm

- ***Algorithm’s Description***

III. We should vary the variables $x_{N\backslash \mathrm{J}}$ such that the solution remains feasible, i. e. $x_{\mathrm{J}} \geq 0, x_{N\backslash \mathrm{J}} \geq 0$. Suppose that $(h_{N\backslash \mathrm{J}})_k > 0, k \in N\backslash \mathrm{J}$ and we want to increase the only non-basic variable $(x_{N\backslash \mathrm{J}})_k = t > 0$. Then the basic variables $x_{\mathrm{J}}(t)$, $x_{\mathrm{J}}(0) = A_J^{-1}b$ will change in the following way

$$x_J(t) = A_J^{-1}b - A_J^{-1}a_k t \geq 0 \quad \Leftrightarrow \quad A_J^{-1}b \geq A_J^{-1}a_k t.$$

- If $(A_J^{-1}a_k)_i \leq 0$ for all $i \in J$ then the feasibility condition is satisfied for all $t > 0$, so the considered LP problem has no solution.
- If the set $\mathcal{I} := \left\{ i \in J \middle| \left(A_J^{-1}a_k\right)_i > 0 \right\} \neq \emptyset$ is not empty then the maximal available $t > 0$ is given by the following formula

$$t^* = \min\left\{\frac{(A_J^{-1}b)_i}{(A_J^{-1}a_k)_i} \mid i \in \mathcal{I}\right\} = \frac{(A_J^{-1}b)_{i^*}}{(A_J^{-1}a_k)_{i^*}} \quad \text{where } i^* \in J.$$

The corresponding point $x_J = x_J(t^*), (x_{N\backslash \mathrm{J}})_k = t^*, (x_{N\backslash \mathrm{J}})_l = 0, l \in N\backslash(\mathrm{J} \cup \{k\})$ is our next basic feasible solution. One can show that the corresponding basis for this feasible solution is $J_{\text{new}} = (\mathrm{J}\backslash i^*) \cup \{k\}$. Now with this new basis we go to the step II.

II. Dantzig’s Simplex Algorithm

• *Algorithm’s Description*

Some remarks:

- During the step III we choose the *leaving variable* from the current basis and the *entering variable* to the new basis from non-basic variables. This is called a *pivot operation*.

- One can show that this operation corresponds to moving from a vertex to another neighboring vertex.

- As you see there can be flexibility of the choice of the indices k and i^* in the step III. This choice is regulated by so-called *pivot rules*. Some of them can lead to cycling of the algorithm.

II. Dantzig's Simplex Algorithm

• *Algorithm's Description*

Below is the equivalent algorithm for the LP problem in the form $\max\{c \cdot x |\, Ax \leq b\}$. Here we suppose that $x_0 \in P$ is an initial vertex of the polyhedron P. Also assume that the inequalities of the system $Ax \leq b$ are ordered as follows: $\alpha_1 x \leq \beta_1, \dots, \alpha_m x \leq \beta_m$.

I. Choose the subsystem $A_0 x \leq b_0$ of the system $Ax \leq b$ such that $A_0 x_0 = b_0$ and the matrix A_0 is invertible.
II. Determine the unique row-vector u such that $c = uA$ and whose components corresponding to rows not belonging to the submatrix A_0 are zero.
III. There are two cases:
   A. $u \geq 0$: then x_0 is an optimal solution.
   B. $u < 0$: choose the minimal index i^* such that $u_{i^*} < 0$. Let y be the vector such that $\alpha y = 0$ for each row α of the matrix A_0 not equal to α_{i^*}, and $\alpha_{i^*} y = -1$. There two cases:
      a) $\alpha y \leq 0$ for each row α of the matrix A. Then the maximum is unbounded.
      b) $\alpha y > 0$ for some row α of the matrix A. Denote by λ_0 the maximal number λ for which the vector $x_0 + \lambda y$ belongs to X. Consider the vector $x_1 = x_0 + \lambda_0 y$. Let j^* be the minimal index for which we have $a_{j^*} x_1 = b_{j^*}$. Also consider the matrix A_1 coinciding with the matrix A_O except the row α_{i^*}, this row is replaced by the row α_{j^*}. Go to the step (I) replacing A_0, x_0 by A_1, x_1.

II. Dantzig’s Simplex Algorithm

• *Simplex Tableau*

Suppose that we want to solve the following problem

$$c \cdot x \to max, x \geq 0, Ax \leq b.$$

To each vertex of the polyhedron $P = \{x \mid x \geq 0, Ax \leq b\} \subset \mathbb{R}^n$ we can associate a table of the following form

$$\begin{array}{c|c} u & \delta \\ \hline D & f \end{array}$$

where u is a row $(n + m)$-vector, D is a $m \times (n + m)$-matrix, f is a column n-vector and δ is a number. Starting with an initial table we will repeat some operation over this table until the vector u becomes non-negative. The final table will correspond to the optimal solution of the problem.

For example, if we have $b \geq 0$ then the vector $x_0 = (0, \ldots, 0)$ is the vertex of P. The corresponding table looks as follows

$$\begin{array}{cc|c} -c & 0 & 0 \\ \hline A & I & b \end{array}$$

II. Dantzig’s Simplex Algorithm

- ***Simplex Tableau***

All tables corresponding to the vertices satisfy the following conditions

$$\begin{array}{c|c} u & \delta \\ \hline D & f \end{array}$$

- $f \geq 0$;
- For each $t = 1, \dots, m$ there is a column of the matrix $\binom{u}{D}$ with index $\sigma(t)$ such that it is the unit basis vector with the unit in the t’th row of the matrix D.

Note that the initial table (for the case $b \geq 0$) $\begin{array}{cc|c} -c & 0 & 0 \\ \hline A & I & b \end{array}$ satisfies these properties.

The operation which will be described on the next slide respects these properties.

II. Dantzig's Simplex Algorithm

- ***Pivot Operation***

The operation for the table $\begin{array}{c|c} u & \sigma \\ \hline D & f \end{array}$ consists of the following steps

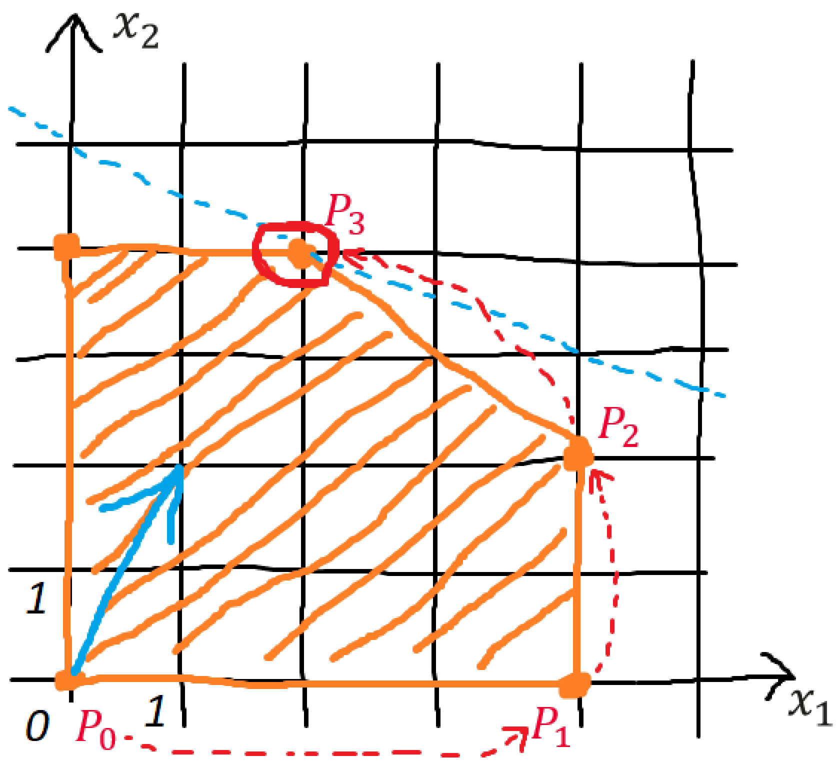
1. Choose the minimal index j^* for which the component $u_{j^*} < 0$;
2. Choose the index t^* such that $D_{t^* j^*} > 0,\ \frac{f_{t^*}}{D_{t^* j^*}} = \min\left\{\frac{f_t}{D_{t j^*}}, t = 1, \dots, m, D_{t j^*} > 0\right\}$ and the index $\sigma(t^*)$ is minimal (*Bland's rule*);
3. Divide the t^*'th row of the table by $D_{t^* j^*}$ and add multiple of this row to other rows in order to obtain the unit basis vector at the j^* column with the unit at the t^*'th row.

If we can't do this operation over the table then this table corresponds to the optimal solution or the problem is unsolvable.

Note that there are other rules for choosing indices (step 1-2), but some of them may cause cycling.

II. Dantzig's Simplex Algorithm

- ***Example I***



Consider the following simple 2-dimensional problem

$$c \cdot x = x_1 + 2x_2 \to \max$$

$$\begin{cases} x_1 \leq 4 \\ x_1 + x_2 \leq 6 \\ x_2 \leq 4 \\ x_1, x_2 \geq 0 \end{cases} \quad \text{equivalent to} \quad \begin{cases} x_1 + \widetilde{x_1} = 4 \\ x_1 + x_2 + \widetilde{x_2} = 6 \\ x_2 + \widetilde{x_3} = 4 \\ x_1, x_2, \widetilde{x_1}, \widetilde{x_2}, \widetilde{x_3} \geq 0 \end{cases}$$

The initial simplex tableau is

P_0

-1	-2	0	0	0	0	
$\boxed{1}$	0	1	0	0	4	$\widetilde{x_1}$
1	1	0	1	0	6	$\widetilde{x_2}$
0	1	0	0	1	4	$\widetilde{x_3}$

. The sequential application of the described operation gives the following tables.

P_1

0	-2	1	0	0	4	
1	0	1	0	0	4	x_1
0	$\boxed{1}$	-1	1	0	2	$\widetilde{x_2}$
0	1	0	0	1	4	$\widetilde{x_3}$

$\to$

P_2

0	0	-1	2	0	6	
1	0	1	0	0	4	x_1
0	1	-1	1	0	2	x_2
0	0	$\boxed{1}$	-1	1	2	$\widetilde{x_3}$

$\to$

Finish!

P_3

0	0	0	1	1	8	
1	0	0	1	-1	2	x_1
0	1	0	0	1	4	x_2
0	0	1	-1	1	2	$\widetilde{x_1}$

II. Dantzig's Simplex Algorithm

- ***Simplex Initialization***

If the vector b contains negative components then the zero-point does not belong to the polyhedron P, so we should choose an initial point in some other way. This procedure is called *simplex initialization*. There are several approaches to do this.

Some of them are based on considering the auxiliary LP problem. One can rewrite the inequality $Ax \leq b$ as two inequalities: $A_1x \leq b_1$ and $A_2x \geq b_2$, where $b_1 \geq 0,\ b_2 > 0$. Then consider the following LP problem

$$\mathbf{1} \cdot (A_2x - y) \to max,$$
$$x \geq 0, y \geq 0, \qquad A_1x \leq b_1, A_2x - y \leq b_2,$$
$$\mathbf{1} = (1,1, \dots, 1).$$

Note that the zero-point $(x, y) = (0,0)$ can be an initial point for this problem. So we can apply the pivot operations starting with the following simplex table. The x-part of the optimal solution for this problem is feasible point for the original problem. This initialization method together with standard simplex method is called the *2-phase simplex method*.

$-1 \cdot A_2$	1	0	0	0
A_1	0	I	0	b_1
A_2	$-I$	0	I	b_2

II. Dantzig's Simplex Algorithm

• *Simplex Initialization*

The two phases of the previous algorithm one can merge into one procedure by introducing a sufficiently big constant M into the objective function. This method is called the *big-M simplex method.*

More precisely, as above rewrite the inequality $Ax \leq b$ as two inequalities: $A_1 x \leq b_1$ and $A_2 x \geq b_2$, where $b_1 \geq 0,\ b_2 > 0$. Then consider the following LP problem

$$c \cdot x + M \cdot \mathbf{1} \cdot (A_2 x - y) \to max,$$
$$x \geq 0, y \geq 0, \qquad A_1 x \leq b_1, A_2 x - y \leq b_2,$$
$$\mathbf{1} = (1,1,\dots,1), \qquad M \gg 1.$$

One can show that always there exists the big constant M such that the x-part of the optimal solution to this new problem is the optimal solution of the original problem. Also we see that we can start with the zero-point in this new problem.

II. Dantzig's Simplex Algorithm

• *Simplex Initialization*

Also there are simplex initialization methods which do not require introduction new variables. The *Nonfeasible Basis (NFB) method* is one of them.

Assume that the vector b contains negative components. In the NFB method we start with the zero-point which is non feasible point for the original problem. Consider the initial simplex table

$$\begin{array}{cc|c} -c & 0 & 0 \\ \hline A & I & b \end{array}$$

1. Take the first index k for which $b_k < 0$.
2. Take the first index j_0 for which $\alpha_{kj_0} < 0$, where α_{ij} is an element of the matrix $[A \;\; I]$.
3. Find the index i_0 such that $\frac{b_{i_0}}{\alpha_{i_0 j_0}} = \min\left\{\frac{b_i}{\alpha_{ij_0}} \mid b_i \geq 0, \alpha_{ij_0} > 0\right\} \cup \left\{\frac{b_k}{\alpha_{kj_0}}\right\}$.
4. Do the pivot operation for the element $\alpha_{i_0 j_0}$. Go to the step 1.

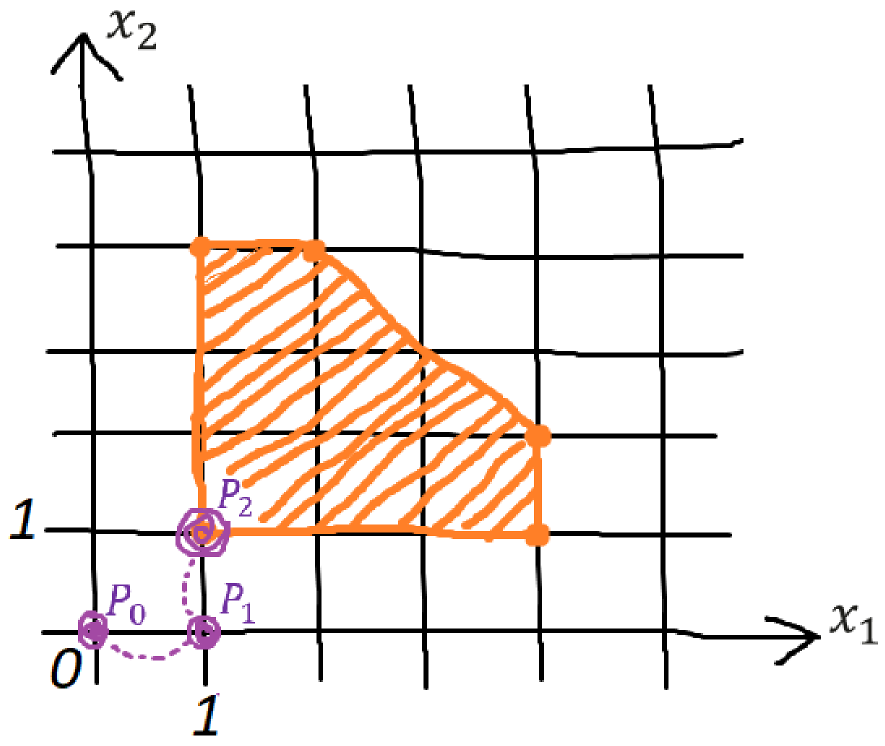
Sequentially applying these steps to the initial simplex table we finally obtain a feasible table with all $b_k \geq 0$. Note that if the step 2 cannot be executed then the corresponding feasible domain is empty and the problem is unsolvable.

II. Dantzig's Simplex Algorithm

- ***Example II***

Consider the example I with two new constraints

$$c \cdot x = x_1 + 2x_2 \to \max, \qquad \begin{cases} x_1 \leq 4 \\ x_1 + x_2 \leq 6 \\ x_2 \leq 4 \\ x_1 \geq 1 \\ x_2 \geq 1 \\ x_1, x_2 \geq 0 \end{cases}$$



Start with the simplex table of the non feasible zero-point. The table for the point P_2 can be used as an initial simplex table for the standard simplex method.

P_0

−1	−2	0	0	0	0	0	0
1	0	1	0	0	0	0	4
1	1	0	1	0	0	0	6
0	1	0	0	1	0	0	4
$\boxed{-1}$	0	0	0	0	1	0	−1
0	−1	0	0	0	0	1	−1

$\to$

P_1

0	−2	0	0	0	−1	0	1
0	0	1	0	0	1	0	3
0	1	0	1	1	0	0	5
0	1	0	0	1	0	0	4
1	0	0	0	0	−1	0	1
0	$\boxed{-1}$	0	0	0	0	1	−1

$\to$

P_2

0	0	0	0	0	−1	−2	3
0	0	1	0	0	1	0	3
0	0	0	1	1	0	1	4
0	0	0	0	1	0	1	3
1	0	0	0	0	−1	0	1
0	1	0	0	0	0	−1	1

III. Duality Theory

- ***Dual LP Problem***

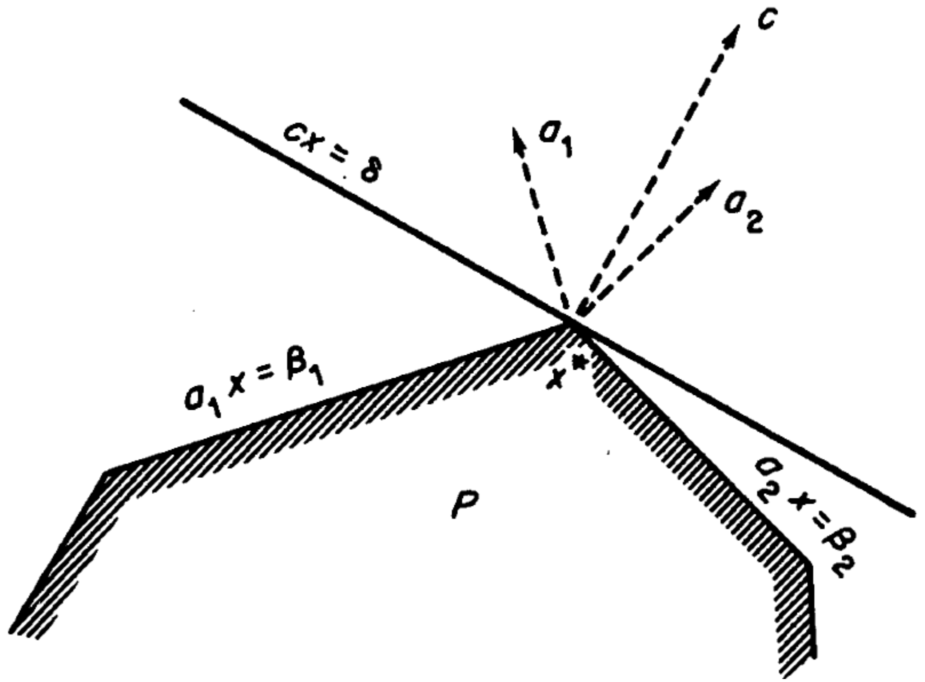
Consider again the following LP problem

$$\max\{c \cdot x \mid Ax \leq b\}.$$

And assume that $x^* \in P$ is an optimal solution of this problem and $c \cdot x^* = \delta$. Also denote the inequalities of the system $Ax \leq b$ which become equalities at the point x^* by $\alpha_1 x \leq \beta_1, \dots, \alpha_n x \leq \beta_n$.

It is easy to see that the equality $c \cdot x = \delta$ can be represented as the linear combination of the equalities $\alpha_1 x = \beta_1, \dots, \alpha_n x = \beta_n$ with non-negative coefficients, i. e. there are numbers $\lambda_1 \geq 0, \dots, \lambda_n \geq 0$ such that

$$c = \lambda_1 \alpha_1 + \dots + \lambda_n \alpha_n,$$
$$\delta = \lambda_1 \beta_1 + \dots + \lambda_n \beta_n.$$



Using these numbers we can construct the row-vector $y \geq 0$ such that we have

$$yA = c, \qquad yb = \delta = c \cdot x^*.$$

In particular, it means that

$$\max\{c \cdot x \mid Ax \leq b\} = \min\{y \cdot b \mid y \geq 0, yA = c\}.$$

The corresponding min-problem on the right hand side is called the *dual LP problem*.

III. Duality Theory

• *Dual LP Problem*

Below is the list of dual problems for *primal* LP problems presented in some different forms

$$\max\{c \cdot x \mid Ax \leq b\} \;\leftrightarrow\; \min\{y \cdot b \mid y \geq 0, yA = c\},$$

$$\max\{c \cdot x \mid x \geq 0, Ax \leq b\} \;\leftrightarrow\; \min\{y \cdot b \mid y \geq 0, yA \geq c\},$$

$$\max\{c \cdot x \mid x \geq 0, Ax = b\} \;\leftrightarrow\; \min\{y \cdot b \mid yA \geq c\}.$$

Weak Duality Theorem. For each feasible solution x^* of the primal problem and each feasible solution of the dual problem y^* we have the inequality

$$c \cdot x^* \leq y^* \cdot b.$$

Strong Duality Theorem. If one of the two problems has an optimal solution then the other one also has an optimal solution and we have the equality

$$\max\{c \cdot x \mid Ax \leq b\} = \min\{y \cdot b \mid y \geq 0, yA = c\}.$$

II. Duality Theory

• *Dual Simplex Algorithm*

Consider the usual LP problem

$$\max\{c \cdot x \mid x \geq 0, Ax \leq b\}$$

and assume that the row-vector c has only non-positive components, i. e. $c \leq 0$. Then we can apply the so-called *dual simplex algorithm* to this problem which works a follows.

Suppose that we have a simplex table $\begin{array}{c|c} u & \delta \\ \hline D & f \end{array}$ which satisfies the following conditions

- all $u_j \geq 0$, but some f_i can be negative;
- for each $t = 1, \dots, m$ there is a column of the matrix $\binom{u}{D}$ with index $\sigma(t)$ such that it is the unit basis vector with the unit in the t'th row of the matrix D.

II. Duality Theory

• *Dual Simplex Algorithm*

We can apply to this simplex table the following dual pivot operation

1. Choose the minimal index i^* for which the component $f_{i^*} < 0$;
2. Choose the index j^* such that $D_{i^*j^*} < 0$ and $\frac{u_{j^*}}{D_{i^*j^*}} = \max\left\{\frac{u_t}{D_{i^*t}}, t = 1, \ldots, m, D_{i^*t} < 0\right\}$;
3. Divide the t^*'th row of the table by $D_{t^*j^*}$ and add multiple of this row to other rows in order to obtain the unit basis vector at the j^* column with the unit at the t^*'th row.

This operation respects the properties of the previous slide. Moreover, if we can't do this operation over the simplex table then this table corresponds to the optimal solution or the problem is unsolvable.

In fact, this dual pivot operation is the primal pivot operation for the dual problem.

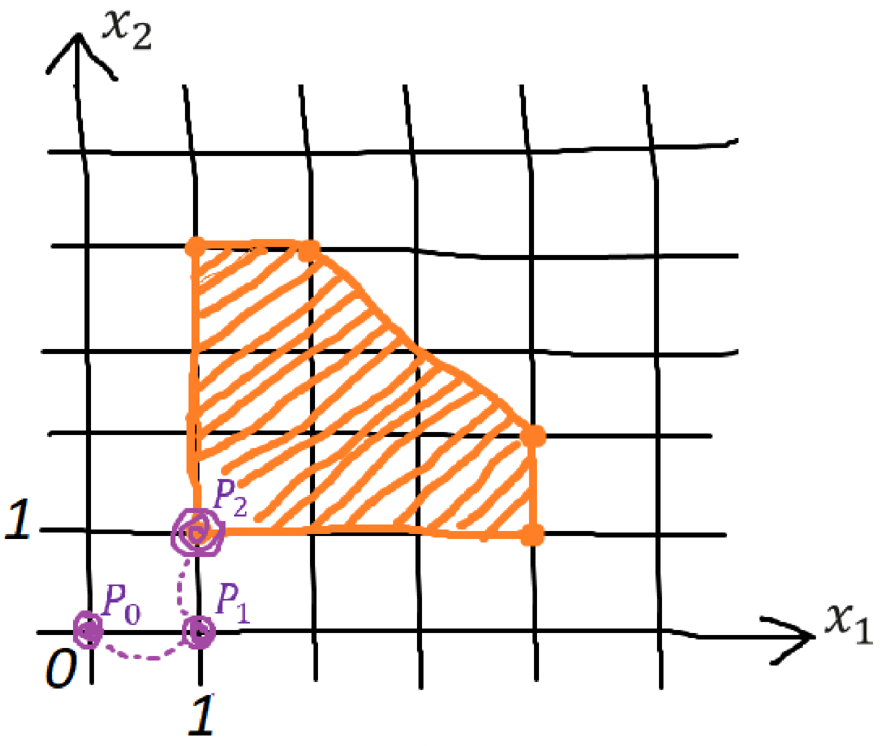
II. Duality Theory

- ***Example III***

Consider the example I with two new constraints

$$x_1 + 2x_2 \to \min \quad \Leftrightarrow \quad -x_1 - 2x_2 \to \max$$

$$\begin{cases} x_1 \leq 4 \\ x_1 + x_2 \leq 6 \\ x_2 \leq 4 \\ x_1 \geq 1 \\ x_2 \geq 1 \end{cases} \quad \Leftrightarrow \quad \begin{cases} x_1 \leq 4 \\ x_1 + x_2 \leq 6 \\ x_2 \leq 4 \\ -x_1 \leq -1 \\ -x_2 \leq -1 \end{cases}$$



Start with the simplex table of the non feasible zero-point. Note that this table satisfies the properties of the dual simplex method.

P_0

1	2	0	0	0	0	0	0
1	0	1	0	0	0	0	4
1	1	0	1	0	0	0	6
0	1	0	0	1	0	0	4
$\boxed{-1}$	0	0	0	0	1	0	−1
0	−1	0	0	0	0	1	−1

$\rightarrow$

P_1

0	2	0	0	0	1	0	−1
0	0	1	0	0	1	0	3
0	1	0	1	1	0	0	5
0	1	0	0	1	0	0	4
1	0	0	0	0	−1	0	1
0	$\boxed{-1}$	0	0	0	0	1	−1

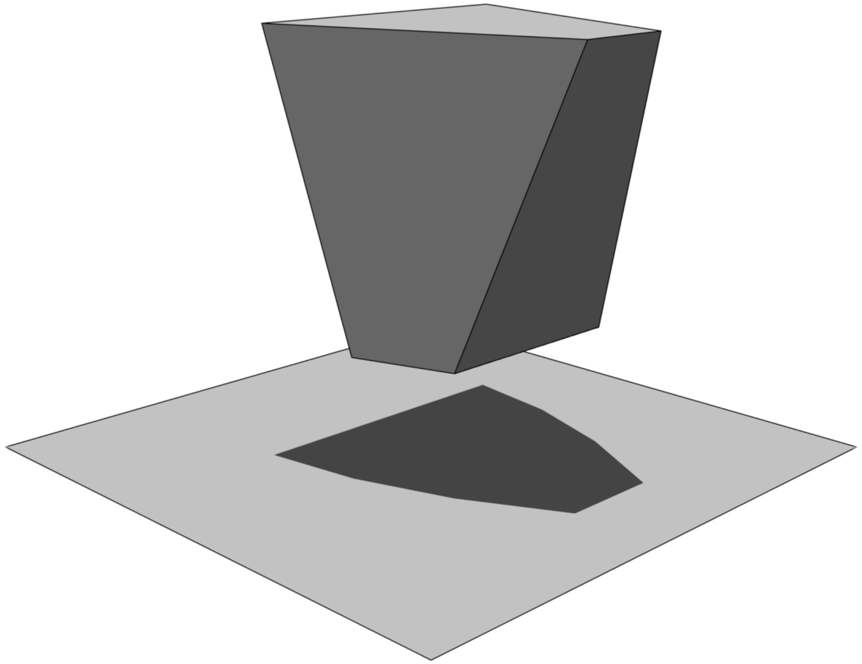
$\rightarrow$

P_2

0	0	0	0	0	1	2	−3
0	0	1	0	0	1	0	3
0	0	0	1	1	0	1	4
0	0	0	0	1	0	1	3
1	0	0	0	0	−1	0	1
0	1	0	0	0	0	−1	1

IV. Complexity of Simplex Algorithm

- ***Exponential Time***



In 1972, Klee and Minty gave an example, the Klee–Minty cube, showing that the *worst-case* complexity of the simplex method as formulated by Dantzig is *exponential time*. However, it is unknown whether there are pivot rules which make the simplex algorithm polynomial.

Analyzing and quantifying the observation that the simplex algorithm is efficient in practice despite its exponential worst-case complexity has led to the development of other measures of complexity. The simplex algorithm has *polynomial-time average-case complexity* under various probability distributions.

$$
\begin{cases}
\max\ 2^{n-1}\zeta_1 + 2^{n-2}\zeta_2 + \dots + 2\zeta_{n-1} + \zeta_n \\
\zeta_1 \le 5 \\
4\zeta_1 + \zeta_2 \le 25 \\
\vdots \\
2^n\zeta_1 + 2^{n-1}\zeta_2 + \dots + 4\zeta_{n-1} + \zeta_n \le 5^n \\
\zeta_1, \dots, \zeta_n \ge 0.
\end{cases}
$$

IV. Complexity of Simplex Algorithm

- ***Solvers***

…

Thank you.

Bring digital to every person, home, and organization for a fully connected, intelligent world.

HUAWEI