CO343 - Operations Research (60016)

Lecture 1

Operations research is the science of taking decisions, it's a branch of applied mathematics where we attempt to model problems where need to make a decision. The decisions aren't arbitrary, and we want to attempt to score each decision based on some metric (such as time, cost, etc.), to find the optimal solution.

The course focuses on formulating a mathematical model to represent the problem, and then developing a computer-based procedure for deriving solutions to the problem from the model. Assume our goal was the following;

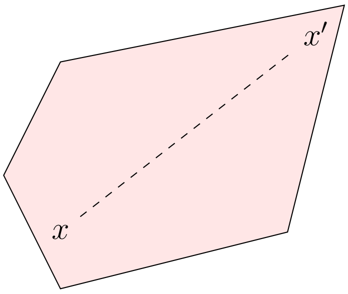
$$\min_{\boldsymbol{x}} z = f(\boldsymbol{x}) \qquad \text{subject to } \boldsymbol{x} \in \mathcal{X}$$

- **decision variables** $\boldsymbol{x} \in \mathbb{R}^n$
- **objective function** $f : \mathbb{R}^n \to \mathbb{R}$
- **feasible set** (set of admissible decisions) $\mathcal{X} \subseteq \mathbb{R}^n$
- **optimal solution** (any vector that minimises f) $\boldsymbol{x}^*$
- **optimal value** $z^* = f(\boldsymbol{x}^*)$

Linear Programming

A linear program optimises a **linear objective function**, where a feasible set is described by linear equality / inequality constraints. Compared to non-linear problems, where a **local** maximum may vary (and therefore be sub-optimal) depending on the starting search position, this isn't a concern for linear problems.

We can say the polygon representing a two dimensional feasible set is convex if the points on the line joining two points in the feasible set are also in the polygon. If this region is convex and linear, it can be proven that a local optimum is also a global optimum. For example, take x and x';



Linear Programming Example

A manufacturer produces A (acid) and C (caustic) and wants to decide a production plan. The ingredients for A and C are X (a sulphate) and Y (sodium).

- each ton of A requires 2 tons of X and 1 ton of Y
- each ton of C requires 1 ton of X and 3 tons of Y
- supply of X is limited to 11 tons per week
- supply of Y is limited to 18 tons per week
- A sells for £1000 per ton

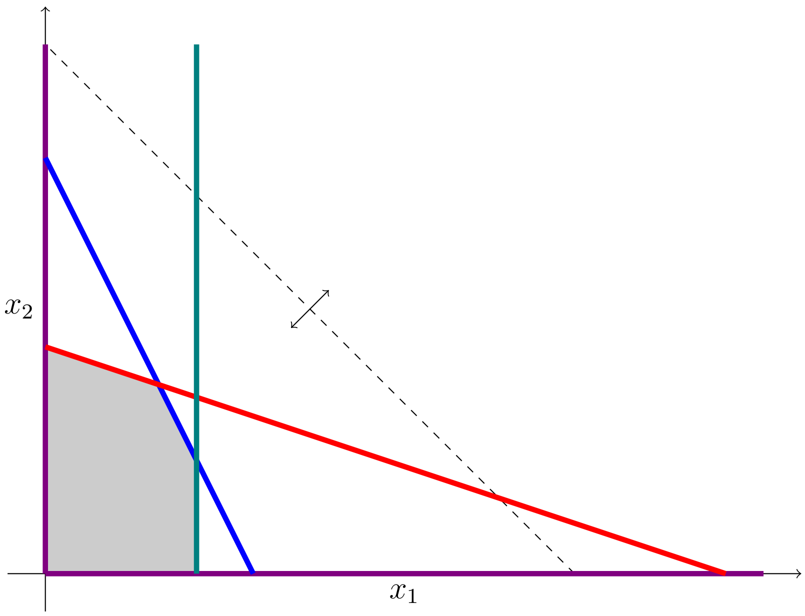
- C sells for £1000 per ton
- a maximum of 4 tons of A can be sold per week

Our goal is to maximise weekly value of sales of A and C. To determine how much A and C to produce, we need to formulate a **mathematical programming model**;

- **decision variables**
  - weekly production of A (tons) x_1
  - weekly production of B (tons) x_2
- **objective function** (weekly profit in £1000s) $z = f(x_1, x_2)$
- **feasible set** $\boldsymbol{x} = (x_1, x_2) \in \mathcal{X}$

A **production plan** is representable as $\boldsymbol{x} = (x_1, x_2)$. The objective function can be written as $z = x_1 + x_2$. Another constraint is that $x_1 \geq 0$ and $x_2 \geq 0$; we cannot produce a negative amount of a product. x_1 tons of A and x_2 tons of C requires $2x_1 + x_2$ tons of X, and we know that is limited to 11 tons per week; therefore we have the constraint $2x_1 + x_2 \leq 11$. Similarly, we also have the limitation of $x_1 + 3x_2 \leq 18$, because of the limitations of Y. Finally, we have another restriction that we cannot sell more than 4 tons of A, therefore $x_1 \leq 4$.

To get the overall feasible set, we intersect the feasible set of all the constraints to get the following;



Each of the following vertices is the intersection of constraints, which can be obtained by solving the linear equation of each line;

$$O = (0, 0)$$
$$P = (0, 6)$$
$$Q = (3, 5)$$
$$R = (4, 3)$$
$$S = (4, 0)$$

By moving the objective function (the dashed line), in the direction of the arrows, we can see that the z value increases further away from the origin, and therefore the graphical result that results in the highest value is Q. Typically the optimal solution lies on a vertex, however in some cases, there can be multiple solutions (an edge when the objective function is parallel to the constraint, or all the points in the feasible set in the case of a constant objective function).

The simplest algorithm is to enumerate all the vertices (intersections) of the feasible set, however this can have exponential complexity in the worst case and the number of vertices grow quite quickly in higher dimensions. The **Simplex Algorithm** finds an optimal vertex, often inspecting a **small subset** of the total.

We can vary this example, for example if we wanted to minimise $z = 3x_1 - x_2$ over the feasible set, we can examine the objective function at each of the vertices;

$O = (0,0)$	$P = (0,6)$	$Q = (3,5)$	$R = (4,3)$	$S = (4,0)$
0	-6	4	9	12

This therefore gives us $P = (x_1, x_2) = (0, 6)$ as the optimal.

On the other hand, if we were to maximise $z = 2x_1 + x_2$, any point on the line segment QR would be optimal; this tells us that points other than the vertices can be optimal, but there is at least one optimal vertex.

Additionally, if we were to set a production goal of 7 tons of A, we'd have an empty feasible set, since $x_1 \geq 7$ would cause an empty set with $x_1 \leq 4$. In this case, the LP is **infeasible**. Similarly, if the constraints on X and Y were removed, the objective function could grow to $+\infty$, hence the LP is **unbounded**.

Lecture 2

Standard Form

In order to use a computer to solve an LP problem, we need to define a **standard form**;

- the goal is to **minimise** a **linear** objective function
- all constraints are linear equality constraints
- all constraint right hand sides are non-negative
- all decision variables are non-negative

A linear problem in standard form is as follows;

$$\begin{array}{lccccccccc} \text{minimise} & z = c_1x_1 & + & c_2x_2 & + & \cdots & c_nx_n & & \\ \\ \text{subject to} & a_{1,1}x_1 & + & a_{1,2}x_2 & + & \cdots & a_{1,n}x_n & = & b_1 \\ & a_{2,1}x_1 & + & a_{2,2}x_2 & + & \cdots & a_{2,n}x_n & = & b_2 \\ & \vdots & & \vdots & & & \vdots & & \vdots \\ & a_{m,1}x_1 & + & a_{m,2}x_2 & + & \cdots & a_{m,n}x_n & = & b_m \end{array}$$

This has the constraints that all decision variables $\forall i \in [1, n]$ $x_i \geq 0$ and $\forall i \in [1, m]$ $b_i \geq 0$. The **input parameters** b_i, c_j, and $a_{i,j}$ are fixed real constants. Clearly, this can be written more compactly as the following;

$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

Therefore, the equation can be written as;

$$\text{minimise } \boldsymbol{z} = \boldsymbol{c}^\top \boldsymbol{x} \text{ subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Note that $\boldsymbol{x} \geq 0$ and $\boldsymbol{b} \geq 0$, which means that it holds **component-wise** (such that $\forall x_i \in \boldsymbol{x}\ x_i \geq 0$).

Standardising

This follows the example in tutorial 1.

Our goal is to maximise $y = 2x_1 + x_2$, (s.t.) subject to;

- $x_1 - 4x_2 \leq 1$
- $-x_1 - 5x_2 \leq -3$
- $x_1, x_2 \geq 0$

We can do the following conversion steps to get the equations into the standard form. To reformulate inequalities as equalities, we introduced the **slack variables** s_1 and s_2. All that is left to do is to convert the maximisation into a minimisation, which can be done by negating the objective function.

$$\begin{aligned}
x_1 - 4x_2 &\leq 1 && \Rightarrow \\
x_1 - 4x_2 + s_1 &= 1 \\
-x_1 - 5x_2 &\leq 3 && \Rightarrow \\
x_1 + 5x_2 &\geq -3 && \Rightarrow \\
x_1 + 5x_2 - s_2 &= -3 \\
x_1, x_2, s_1, s_2 &\geq 0 \\
\text{(maximise) } y &= 2x_1 + x_2 && \Rightarrow \\
\text{(minimise) } z &= -2x_1 - x_2
\end{aligned}$$

Therefore, we can therefore say a minimisation of $\boldsymbol{z} = \boldsymbol{c}^\top \boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ and $\boldsymbol{x} \geq 0$ is equivalent to the same minimisation subject to $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{b}$ and $\boldsymbol{x}, \boldsymbol{s} \geq 0$. The slack variables take the value of the difference $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$. Similarly, **excess variables** are the same, but instead of being added to the left hand side of the inequality, they are subtracted, and therefore take the value of the difference $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$. Additionally, a change of sign for the right hand side is trivial, as it can be done by multiplying the entire inequality by -1.

Free Variables

Suppose the constraint $x_j \geq 0$ does not exist, such that it can be positive or negative. We can do this by substituting $x_j = x_j^+ - x_j^-$. The LP now has the following $n + 1$ variables;

$$x_1, \dots, x_{j-1}, x_j^+, x_j^-, x_{j+1}, \dots, x_n$$

Another approach to introduce free variables is to use substitution. Any **equality constraint** involving x_j can be used to eliminate x_j, as for x_1 in the following conditions (with the substitution of $x_1 = 5 - 3x_2 - x_3$);

$$\begin{aligned} &\text{(minimise) } z = x_1 + 3x_2 + 4x_3 \\ &x_1 + 2x_2 + x_3 = 5 \\ &2x_1 + 3x_2 + x_3 = 6 \qquad \Rightarrow \end{aligned}$$

$$\begin{aligned} &\text{(minimise) } z = x_2 + 3x_3 + 5 \\ &x_2 + x_3 = 4 \end{aligned}$$

Tutorial

2. A company produces laptops at two factories, A and B. In factory A, s_A laptops are produced a year, and s_B laptops are produced a year in factory B. The three stores, 1, 2, and 3, sell d_1, d_2, and d_3 a year. The cost of shipping a laptop from the factory $i \in \{A, B\}$ to store $j \in \{1, 2, 3, \}$ is $c_{i,j}$. Assume that the demand of all stores can be satisfied, such that $s_A + s_B \geq d_1 + d_2 + d_3$.

   1. How should the laptops be shipped from the two factories to minimise shipping costs, assuming the following;

      $$\begin{bmatrix} s_A \\ s_B \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

      $$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

      $$(c_{i,j}) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix} \qquad \text{(first row corresponds to store } A\text{)}$$

   2. Formulate the optimisation model corresponding to the previous question, using the general parameters;

      Note that we will denote the number of laptops from each factory $i \in \{A, B\}$ to store $j \in \{1, 2, 3\}$ as $x_{i,j}$. We therefore want to minimise the following;

      $$z = \sum_i \sum_j c_{i,j} x_{i,j}$$

      Under the following conditions;

      $$\begin{aligned} x_{A,j} + x_{B,j} &= d_j & \forall j \in \{1, 2, 3\} \\ x_{i,1} + x_{i,2} + x_{i,3} &\leq s_i & \forall i \in \{A, B\} \\ x_{i,j} &\geq 0 & \forall i, \forall j \end{aligned}$$

      It's important to note that satisfying demand is to use equality, as we can reduce the amount of computation we need to do.

Lecture 3

We now only focus on LPs in **standard form**; minimise $z = \boldsymbol{c}^\top \boldsymbol{x}$, subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} \geq 0$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m \geq 0, c \in \mathbb{R}^n$. We also assume that (the number of variables) $n \geq m$ (the number of equations), otherwise the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is overdetermined. Similarly, we also assume that the rows of $\boldsymbol{A}$ are linearly independent, otherwise constraints are redundant or consistent. Therefore, we can say $\text{rk}(\boldsymbol{A}) = m$. If there is linear dependence, we have either;

- **contradictory constraints** (no solution)

$$\begin{aligned} x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned}$$

- **redundant constraints**

$$\begin{aligned} x_1 + x_2 &= 1 \\ 2x_1 + 2x_2 &= 2 \end{aligned}$$

For now, we focus only on the system of linear equations in $\mathcal{LP}$;

$$\boldsymbol{Ax} = \boldsymbol{b}$$

Let $\boldsymbol{A} = [\boldsymbol{a_1}, \dots, \boldsymbol{a_n}]$ where $a_i \in \mathbb{R}^m$ is the i^{th} column vector of $\boldsymbol{A}$. We want to select a subset of m columns $\boldsymbol{a_i}$ that are linearly independent - which will always be possible since $n \geq m = \text{rk}(\boldsymbol{A})$. This gives us a square matrix for us to solve. The **index set** I consists of the indices for those m columns, hence $I \subseteq \{1, \dots, n\}$. We define the matrix $\boldsymbol{B} = B(I) \in \mathbb{R}^{m \times m}$ consisting of the columns $\{\boldsymbol{a_i}\}_{i \in I}$ as the **basis** corresponding to the index set I.

We define a solution $\boldsymbol{x}$ to $\boldsymbol{Ax} = \boldsymbol{b}$ with $\forall i \notin I$ $(x_i = 0)$ as a **basic solution (BS)** to $\boldsymbol{Ax} = \boldsymbol{b}$ with respect to the index set I. Similarly, we define a solution $\boldsymbol{x}$ satisfying both $\boldsymbol{Ax} = \boldsymbol{b}$ and $\boldsymbol{x} \geq 0$ as a **feasible solution (FS)**. A feasible solution, which is also basic, is a **basic feasible solution (BFS)**.

Assume, for the example $I = \{1, \dots, m\}$.

$$\begin{array}{ccccccccccc} a_{1,1}x_1 & + & \dots & + & a_{1,m}x_m & + & a_{1,m+1}x_{m+1} & + & \dots & + & a_{1,n}x_n & = & b_1 \\ a_{2,1}x_1 & + & \dots & + & a_{2,m}x_m & + & a_{2,m+1}x_{m+1} & + & \dots & + & a_{2,n}x_n & = & b_2 \\ \vdots & & & & \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m,1}x_1 & + & \dots & + & a_{m,m}x_m & + & a_{m,m+1}x_{m+1} & + & \dots & + & a_{m,n}x_n & = & b_m \end{array}$$

This is then equivalent to $\boldsymbol{Bx_B} = \boldsymbol{b}$;

$$\begin{array}{ccccccccccccc} a_{1,1}x_1 & + & \dots & + & a_{1,m}x_m & + & a_{1,m+1}0 & + & \dots & + & a_{1,n}0 & = & b_1 \\ a_{2,1}x_1 & + & \dots & + & a_{2,m}x_m & + & a_{2,m+1}0 & + & \dots & + & a_{2,n}0 & = & b_2 \\ \vdots & & & & \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m,1}x_1 & + & \dots & + & a_{m,m}x_m & + & a_{m,m+1}0 & + & \dots & + & a_{m,n}0 & = & b_m \end{array}$$

By removing the 0 terms, we can simplify it to the following;

$$\begin{array}{ccccccc} a_{1,1}x_1 & + & \dots & + & a_{1,m}x_m & = & b_1 \\ a_{2,1}x_1 & + & \dots & + & a_{2,m}x_m & = & b_2 \\ \vdots & & & & \vdots & & \vdots \\ a_{m,1}x_1 & + & \dots & + & a_{m,m}x_m & = & b_m \end{array}$$

We can observe that the **basic solution** corresponding to I is unique, since the vectors $\{\boldsymbol{a_i}\}_{i \in I}$ are linearly independent, the basis $\boldsymbol{B}$ is invertible, and has the following unique solution;

$$\boldsymbol{x_B} = \boldsymbol{B}^{-1}\boldsymbol{b} \in \mathbb{R}^m$$

Therefore, we can define the vector $\boldsymbol{x}$ as;

$$x_i = \begin{cases} \boldsymbol{x_B}_i & i \in I \\ 0 & i \notin I \end{cases}$$

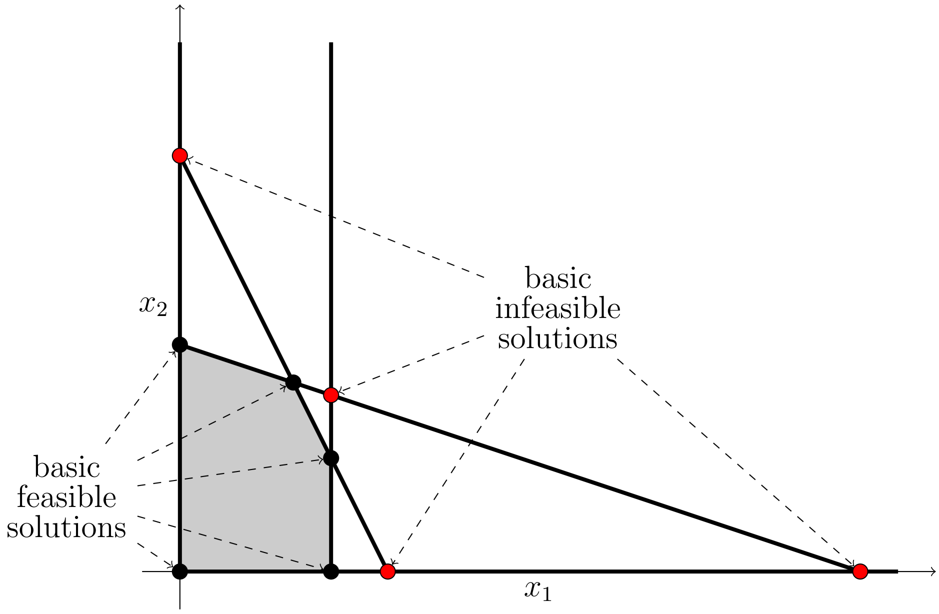
This $\boldsymbol{x}$ is the **unique basic solution** to $\boldsymbol{Ax} = \boldsymbol{b}$ with respect to I. However - this doesn't mean it's feasible, as we could end up with negative values. The geometric intuition that the corners of the feasible set correspond are LP come back into play, when we consider that the corners of the feasible set correspond to **basic feasible solutions**.

Consider the example from the first lecture (note that each line in the previously drawn graph denotes when a variable in the standard form is zero);

$$\begin{aligned} y &= x_1 + x_2 && \text{objective function} \\ 2x_1 + x_2 &\leq 11 && \text{constraint on availability of } X \\ x_1 + 3x_2 &\leq 18 && \text{constraint on availability of } Y \\ x_1 &\leq 4 && \text{constraint on demand of } A \\ x_1, x_2 &\geq 0 && \text{non-negativity constraints} \end{aligned}$$

In standard form:

$$\begin{aligned} n &= 5 && \text{number of variables} \\ m &= 3 && \text{number of constraints} \\ z &= -x_1 - x_2 && \text{objective function} \\ 2x_1 + x_2 + x_3 &= 11 \\ x_1 + 3x_2 + x_4 &= 18 \\ x_1 + x_5 &= 4 \\ x_1, x_2, x_3, x_4, x_5 &\geq 0 \end{aligned}$$



The intuition is that the vertices of the feasible set are the basic feasible solutions. Therefore, an optimum is always at a vertex in geometry, hence an optimum is always achieved at a **BFS** in algebra.

For an LP in standard form with $\text{rk}(\boldsymbol{A}) = m \leq n$;

1. if there exists a feasible solution, there exists a BFS
2. if there exists an optimal solution, there exists an optimal BFS

However, there may be feasible / optimal solutions that are not BFS.

The first theorem reduces solving a LP to searching over BFS's, there are a finite number of ways to select m columns for I for an LP in standard form with n variables and m constraints;

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

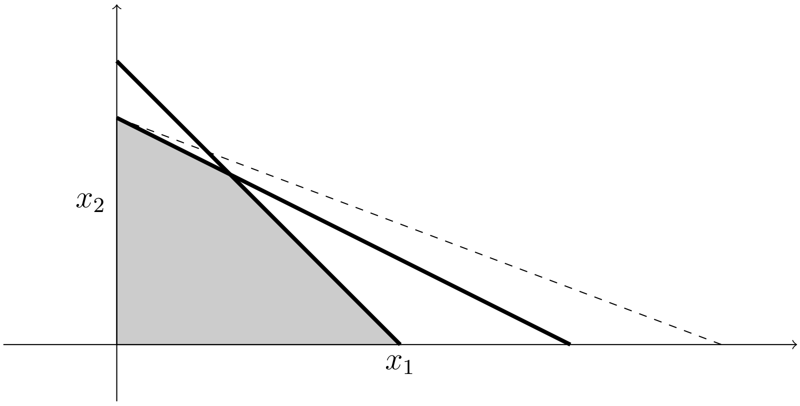
This gives an obvious, but very inefficient, method through a finite search. The number of distinct BFS is usually less than that upper bound however, as $B(I)$ may be singular (non-invertible), or the corresponding BS may not be feasible.

Example

Consider the following optimisation problem;

$$\begin{aligned} \text{maximize} \quad & y = 3x_1 + 4x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 4 \\ & 2x_1 + x_2 \leq 5 \\ & x_1, x_2 \geq 0 \end{aligned}$$

This has the following graphical representation;



We then want to convert this into standard form as follows;

$$\begin{aligned} -\ \text{minimize} \quad & z = -3x_1 - 4x_2 \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 4 \\ & 2x_1 + x_2 + x_4 = 5 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

From this, we have 4 columns, and let us choose our index set $I = \{1, 2\}$. Therefore;

$$\begin{aligned} \boldsymbol{B} &= \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \\ \boldsymbol{B}^{-1} &= \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \\ \boldsymbol{x_B} &= \boldsymbol{B}^{-1}\boldsymbol{b} \\ &= \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 3 \end{bmatrix} \\ \boldsymbol{x} &= \begin{bmatrix} 1 \\ 3 \\ 0 \\ 0 \end{bmatrix} && \text{this is a basic feasible solution} \end{aligned}$$

With a fixed index set I where $|I| = m$ and $B(I)$ invertible. The variables $\{x_i\}_{i \in I}$ are referred to as basic variables, while the other variables $\{x_i\}_{i \notin I}$ are referred to as the nonbasic variables corresponding to I. Nonbasic variables are **always** zero, but the basic variables can be anything (including zero).

The **basic representation** corresponding to I is the unique reformulation of the system $z = \boldsymbol{c}^\top \boldsymbol{x}$ and $\boldsymbol{Ax} = \boldsymbol{b}$, which expresses the objective function value z and each basic variable as a linear function of the nonbasic variables;

$$\begin{bmatrix} z \\ \boldsymbol{x_B} \end{bmatrix} = f(\boldsymbol{x_N})$$

- $\boldsymbol{x_B} = [x_i \mid i \in I]$ (basic variable)
- $\boldsymbol{x_N} = [x_i \mid i \notin I]$ (nonbasic variable)
- $f : \mathbb{R}^{n-m} \to \mathbb{R}^{m+1}$ is **linear**

Once again, let $\boldsymbol{A} = [\boldsymbol{a_1}, \dots, \boldsymbol{a_n}]$. where $\boldsymbol{a_i} \in \mathbb{R}^m$ is the i^{th} column of $\boldsymbol{A}$. Take any index set $I \subseteq \{1, \dots, m\}$ with $|I| = m$, we can define the following;

$$\begin{aligned}
\boldsymbol{B} &= [\boldsymbol{a_i} \mid i \in I] \\
\boldsymbol{N} &= [\boldsymbol{a_i} \mid i \notin I] \\
\boldsymbol{c_B} &= [c_i \mid i \in I] \\
\boldsymbol{c_N} &= [c_i \mid i \notin I] \\
\boldsymbol{x_B} &= [x_i \mid i \in I] \\
\boldsymbol{x_N} &= [x_i \mid i \notin I]
\end{aligned}$$

This implies that;

$$\begin{aligned}
\boldsymbol{Ax} &= \boldsymbol{Bx_B} + \boldsymbol{Nx_N} \\
\boldsymbol{c}^\top \boldsymbol{x} &= \boldsymbol{c_B}^\top \boldsymbol{x_B} + \boldsymbol{c_N}^\top \boldsymbol{x_N}
\end{aligned}$$

Given this partition, we have the following;

$$\left.\begin{aligned} z &= \boldsymbol{c}^\top \boldsymbol{x} \\ \boldsymbol{Ax} &= \boldsymbol{b} \end{aligned}\right\} \Leftrightarrow \left\{\begin{aligned} z &= \boldsymbol{c_B}^\top \boldsymbol{x_B} + \boldsymbol{c_N}^\top \boldsymbol{x_N} \\ \boldsymbol{Bx_B} &= \boldsymbol{b} - \boldsymbol{Nx_N} \end{aligned}\right.$$

Since $\boldsymbol{B}$ is invertible, we can get the following;

$$\begin{aligned}
\boldsymbol{x_B} &= \boldsymbol{B}^{-1}(\boldsymbol{b} - \boldsymbol{Nx_N}) \\
&= \boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{Nx_N} \\
z &= \boldsymbol{c_B}^\top \boldsymbol{x_B} + \boldsymbol{c_N}^\top \boldsymbol{x_N} \\
&= \boldsymbol{c_B}^\top (\boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{Nx_N}) + \boldsymbol{c_N}^\top \boldsymbol{x_N} \\
&= \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{b} + (\boldsymbol{c_N}^\top - \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{N})\boldsymbol{x_N} \\
&= \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{b} + (\boldsymbol{c_N} - \boldsymbol{N}^\top \boldsymbol{B}^{-\top}\boldsymbol{c_B})^\top \boldsymbol{x_N} && \text{where } \boldsymbol{B}^{-\top} = (\boldsymbol{B}^{-1})^\top
\end{aligned}$$

Therefore, the basic representation is as follows;

$$\begin{aligned}
z &= \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{b} + (\boldsymbol{c_N} - \boldsymbol{N}^\top \boldsymbol{B}^{-\top}\boldsymbol{c_B})^\top \boldsymbol{x_N} \\
\boldsymbol{x_B} &= \boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{Nx_N}
\end{aligned}$$

This expresses z and $\boldsymbol{x_B}$ as linear functions of $\boldsymbol{x_N}$. However by setting $\boldsymbol{x_N} = \boldsymbol{0}$, we obtain the basic solution $\boldsymbol{x} = (\boldsymbol{x_B}, \boldsymbol{x_N}) = (\boldsymbol{B}^{-1}\boldsymbol{b}, \boldsymbol{0})$, with the objective value $z = \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{b}$. The **reduced cost vector** is $\boldsymbol{r} = \boldsymbol{c_N} - \boldsymbol{N}^\top \boldsymbol{B}^{-\top}\boldsymbol{c_B}$, which characterises the sensitivity of the objective function value z with respect to the nonbasic variables.

Referring back to the previous example;

– minimize $\quad z = -3x_1 - 4x_2$

$$\begin{aligned} \text{subject to} \quad x_1 + x_2 + x_3 &= 4 \\ 2x_1 + x_2 + x_4 &= 5 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

Consider the solution we get when;

$$\begin{aligned} I &= \{3, 4\} \\ O &= (0, 0, 4, 5) \\ z &= -3x_1 - 4x_2 \\ x_3 &= 4 - x_1 - x_2 \\ x_4 &= 5 - 2x_1 - x_2 \end{aligned}$$

However, by looking at the objective function, we can see that it is more desirable to increase the value of x_2, so we fix $x_1 = 0$. This then gives us the following;

$$\begin{aligned} z &= -4x_2 && \\ x_3 &= 4 - x_2 && \\ &\geq 0 && \\ x_4 &= 5 - x_2 && \\ &\geq 0 && \Rightarrow \\ x_2 &\leq 4 && \Rightarrow \\ x_2 &= 4 && \Rightarrow \\ x_3 &= 0 && \end{aligned}$$

This **pivoting** changes the index set to be $I = \{2, 4\}$. Looking at the nonbasic variables $\{x_1, x_3\}$;

$$\begin{aligned} z &= -3x_1 - 4x_2 \\ &= -3x_1 - 4(4 - x_1 - x_3) \\ &= -3x_1 - 16 + 4x_1 + 4x_3 \\ &= -16 + x_1 + 4x_3 \\ x_2 &= 4 - x_1 - x_3 \end{aligned}$$

We can see, by looking at the coefficients, that x_1 and x_3 will cause the minimal solution to increase if they weren't zero.

Lecture 4

Simplex Tableau

If we consider a basic representation of the following form, where the reduced cost vector $\boldsymbol{r} = \boldsymbol{c_N} - \boldsymbol{N}^\top \boldsymbol{B}^{-\top} \boldsymbol{c_B}$;

$$\begin{aligned} z - \boldsymbol{r}^\top \boldsymbol{x_N} &= \boldsymbol{c_B}^\top \boldsymbol{B}^{-1} \boldsymbol{b} \\ \boldsymbol{x_B} + \boldsymbol{B}^{-1} \boldsymbol{N} \boldsymbol{x_N} &= \boldsymbol{B}^{-1} \boldsymbol{b} \end{aligned}$$

We can represent it in the following **tableau**;

BV	z	$\boldsymbol{x_B}^\top$	$\boldsymbol{x_N}^\top$	RHS
z	1	$\boldsymbol{0}^\top$	$-\boldsymbol{r}^\top$	$\boldsymbol{c_B}^\top \boldsymbol{B}^{-1} \boldsymbol{b}$
$\boldsymbol{x_B}$	$\boldsymbol{0}$	$\boldsymbol{I}$	$\boldsymbol{B}^{-1} \boldsymbol{N}$	$\boldsymbol{B}^{-1} \boldsymbol{b}$

Note that here $\boldsymbol{I} \in \mathbb{R}^{m \times m}$ is an identity matrix. Also note the separation of the basic and non-basic variables for the tableau - typically we will simply write it in lexicographical order. If $\boldsymbol{B}^{-1}\boldsymbol{b} \geq 0$ then we can denote it as a BFS.

Consider the previous example;

$$
\begin{aligned}
z &= -3x_1 - 4x_2 \\
x_3 &= 4 - x_1 - x_2 \\
x_4 &= 5 - 2x_1 - x_2
\end{aligned}
$$

Note that the basic variables have a specific property where they are 0s in the columns, other than a 1 in its respective row;

BV	z	x_1	x_2	x_3	x_4	RHS
z	1	3	4	0	0	0
x_3	0	1	1	1	0	4
x_4	0	2	1	0	1	5

Consider the basic representation from the example, with the index set $I = \{1, 2, 5\}$, with the following explicit formulation;

$$
\begin{aligned}
z - \frac{2}{5}x_3 - \frac{1}{5}x_4 &= -8 \\
x_2 - \frac{1}{5}x_3 + \frac{2}{5}x_4 &= 5 \\
-\frac{3}{5}x_3 + \frac{1}{5}x_4 + x_5 &= 1 \\
x_1 + \frac{3}{5}x_3 - \frac{1}{5}x_4 &= 3
\end{aligned}
$$

This can now be set in the tableau as;

BV	z	x_1	x_2	x_3	x_4	x_5	RHS
z	1	0	0	$-\frac{2}{5}$	$-\frac{1}{5}$	0	-8
x_2	0	0	1	$-\frac{1}{5}$	$\frac{2}{5}$	0	5
x_5	0	0	0	$-\frac{3}{5}$	$\frac{1}{5}$	1	1
x_1	0	1	0	$\frac{3}{5}$	$-\frac{1}{5}$	0	3

The tableau is a practical way to analyse the basic solution associated to the basic representation;

- the RHS of the objective row is the objective **value** of the current basic solution
- the RHS's of the other rows are the values of the basic variables at the current basic solution
- the coefficients of the non-basic variables in the **objective row** are the **negative reduced costs**
- the current basic solution is feasible iff all the RHS's are ≥ 0 (but the objective row can be negative)

The general tableau for a feasible index set I, with $p \in I, q \notin I$;

BV	z	x_1	$\cdots$	x_p	$\cdots$	x_q	$\cdots$	x_n	RHS
z	1	β_1	$\cdots$	$\beta_p(= 0)$	$\cdots$	β_q	$\cdots$	β_n	β_0
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\vdots$
x_p	0	$y_{p,1}$	$\cdots$	$y_{p,p}$	$\cdots$	$y_{p,q}$	$\cdots$	$y_{p,n}$	$y_{p,0}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\vdots$

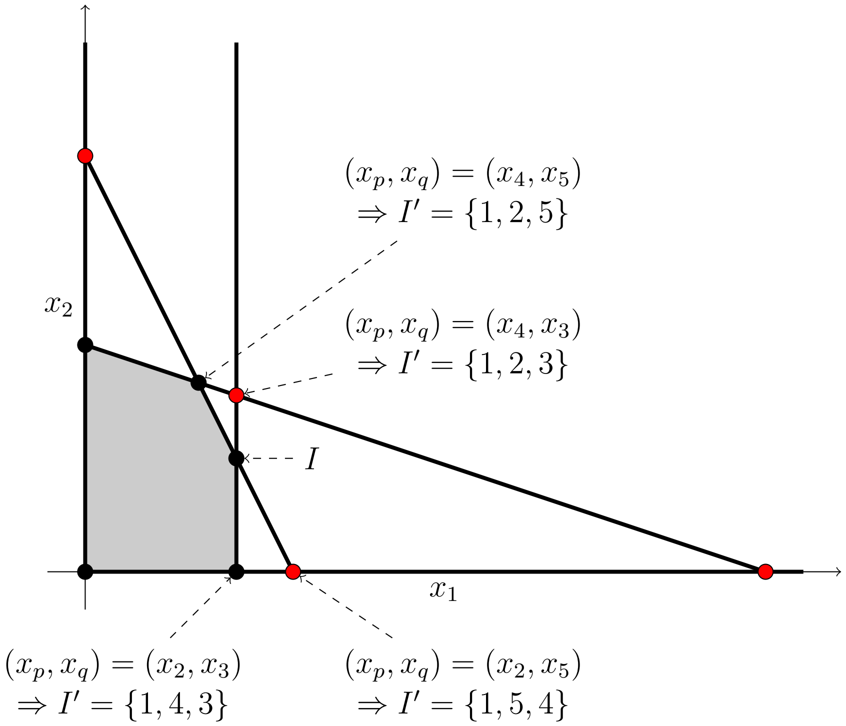
This has the following properties;

- $\forall i \in I \; y_{i,i} = 1$, and $\forall i \in I, j \in I \setminus \{i\} \; y_{j,i} = 0$
- $\forall i \notin I, i \neq 0 \; \beta_i = -r_i$ negative reduced cost
- $\forall i \in I \; \beta_i = 0$

Pivoting

The idea of the Simplex algorithm is that if a vertex x for the index set I is not optimal, then one of its neighbouring vertices will have a **better objective value**. Neighbouring are obtained by swapping a basic variable x_p with a non-basic variable x_q, to obtain a new index set I' - x_p **leaves** the basis and x_q **enters** the basis. The technique called **pivoting** is used to efficiently compute the new basic representation by updating I to I'. This is similar to applying elementary row operations in Gaussian elimination, and the pair (p, q) is referred to as the **pivot**.

While it's possible to pivot to something that isn't feasible, in this algorithm we will only look at pivots to feasible solutions. Consider the following, starting with the basic solution for the index set $I = \{1, 2, 4\}$;


$(x_p, x_q) = (x_4, x_5)$
$\Rightarrow I' = \{1, 2, 5\}$
x_2
$(x_p, x_q) = (x_4, x_3)$
$\Rightarrow I' = \{1, 2, 3\}$
I
x_1
$(x_p, x_q) = (x_2, x_3)$
$\Rightarrow I' = \{1, 4, 3\}$
$(x_p, x_q) = (x_2, x_5)$
$\Rightarrow I' = \{1, 5, 4\}$

In order to swap x_p and x_q we perform the following steps;

1. **divide** row p by the pivot element $y_{p,q}$ and relabel it as row q;

$$\forall j = 0, \dots, n \quad \left[y'_{q,j} = \frac{y_{p,j}}{y_{p,q}} \right]$$

2. **subtract** row p multiplied by $\frac{y_{i,q}}{y_{p,q}}$ from row $i \in I \setminus \{p\}$

$$\forall j = 0, \dots, n \quad \left[y'_{i,j} = y_{i,j} \frac{y_{i,q}}{y_{p,q}} y_{p,j} \right]$$

3. **subtract** row p multiplied by $\frac{\beta_q}{y_{p,q}}$ from the objective row

$$\forall j = 0, \dots, n \quad \left[\beta'_j = \beta_j - \frac{\beta_q}{y_{p,q}} y_{p,j} \right]$$

Applying the following steps to the example table (note that a new horizontal line denotes a new table). Note that we are swapping out x_4 for x_1, hence $y_{p,q} = y_{4,1} = 2$;

BV	z	x_1	x_2	x_3	x_4	RHS
z	1	3	4	0	0	0
x_3	0	1	1	1	0	4
x_4	0	2	1	0	1	5
z	1	0	$\frac{5}{2}$	0	$-\frac{3}{2}$	$-\frac{15}{2}$
x_3	0	0	$\frac{1}{2}$	1	$-\frac{1}{2}$	$\frac{3}{2}$
x_4	0	1	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{5}{2}$

Note that both the RHS's of the basic variables are non-negative, hence we have a BFS. However, since there are positive coefficients for the nonbasic variables in the objective row, we can still improve this value.

We need a way to choose the variable x_q, which enters the basis. Consider that the objective row is equivalent to;

$$z + \sum_{i=1}^{n} \beta_i x_i = \beta_0 \Leftrightarrow z = \beta_0 - \sum_{i \notin I} \beta_i x_i$$

We also know the following, by definition;

$$\beta_i = \begin{cases} 0 & \text{if } i \in I \quad \text{(basic variables)} \\ -r_i & \text{if } i \notin I \quad \text{(nonbasic variables)} \end{cases}$$

Any nonbasic x_i with $\beta_i > 0$ can enter the basis and become x_q, since each of them will decrease z, however, we can use the following steps to choose one;

- if there only exists a single x_i with $\beta_i > 0$, pick this as x_q
- if several x_i have $\beta_i > 0$, pick x_i with **largest** β_i
- if several x_i have the same largest β_i, pick the **smallest** index x_i

Similarly, we also need to choose which basic variable x_p to leave the basis. We need to ensure the following for all variables x_i in the index set I;

$$x_i = y_{i,0} - y_{i,q} x_q \geq 0 \Leftrightarrow \begin{cases} x_q \leq \bar{x}_{i,q} \triangleq \dfrac{y_{i,0}}{y_{i,q}} & \text{if } y_{i,q} > 0 \\ x_q \leq \bar{x}_{i,q} \triangleq \infty & \text{if } y_{i,q} \leq 0 \end{cases}$$

This means that if $y_{i,q}$ is positive, we have an upper bound, however if it's negative (or zero), it is unbounded (hence ∞). For this to be feasible, we want to ensure that x_q is set such that all bounds are simultaneously satisfied;

$$x_q \leq \min_{i \in I} \bar{x}_{i,q}$$

There are two cases for picking the variable x_p to leave the basis;

- **trivial bounds** ($\min_{i \in I} \bar{x}_{i,q} = \infty$)

  In this case, the entering variables x_q can grow indefinitely. Since we have $\beta_q > 0$, the objective value $z = \beta_0 - \beta_q x_q$ can drop indefinitely, hence the LP is unbounded. In this case, we don't need to choose an x_p variable.

- **non-trivial**

  Here the best value of the objective is obtained by maximising x_q, hence setting;

  $$x_q = \min_{i \in I} \bar{x}_{i,q}$$

  We can call p the row such that $\bar{x}_{p,q} = \min_{i \in I} \bar{x}_{i,q}$, which is the row that constraints the most the increase in value of x_q. Similarly, if there are multiple p satisfying this, we can choose the one with the smallest index.

In summary, the simplex algorithm (minimisation) is as follows;

1. find initial BFS and its basic representation
2. if $\beta_i \leq 0$ for all $i \notin I$; we can stop, the current BFS is optimal
3. if $\exists j \notin I$ with $\beta_j > 0$ and $y_{i,j} \leq 0$ for all $i \in I$; we can stop, no finite minimum exists
4. choose x_q with the largest $\beta_q > 0$ (x_q enters the basis)
5. choose $p \in \operatorname{argmin}_{i \in I} \bar{x}_{i,q}$ (x_p leaves the basis)
6. pivot on $y_{p,q}$ and repeat from step 2

In the example below, I will denote the β_i chosen for x_q in violet, the pivot in teal, and calculations for $\bar{x}_{i,q}$ in blue.

BV	z	x_1	x_2	x_3	x_4	RHS
z	1	3	4	0	0	0
x_3	0	1	1	1	0	4 $\frac{4}{1} = \bar{x}_{3,2}$
x_4	0	2	1	0	1	5 $\frac{5}{1} = \bar{x}_{4,2}$
z	1	-1	0	-4	0	-16
x_2	0	1	1	1	0	4
x_4	0	1	0	-1	1	1

Note that both the coefficients are negative in the objective row, we have the following optimal solution;

$$z^* = -16$$
$$y^* = 16$$
$$x^* = (0, 4, 0, 1)$$

Tutorial

2. Consider the following optimisation problem;

$$\begin{aligned} \text{maximize} \quad & y = x_1 + 3x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \leq 4 \\ & x_1 + 2x_2 \leq 4 \\ & x_1, x_2 \geq 0 \end{aligned}$$

(a) Bring the problem into standard form by introducing slack variables s_1 and s_2.

$$\begin{aligned} -\text{ minimize} \quad & z = -x_1 - 3x_2 \\ \text{subject to} \quad & 2x_1 + x_2 + s_1 = 4 \\ & x_1 + 2x_2 + s_2 = 4 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned}$$

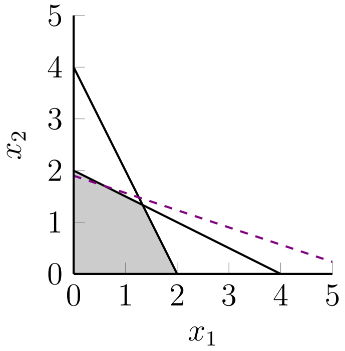
(b) For the problem in standard form, determine all basic solutions. Which of these problems are feasible, and what are their objective values?

$$\begin{aligned} \text{BV} &= \{x_1, x_2\} \\ \text{NBV} &= \{s_1, s_2\} \\ 2x_1 + x_2 &= 4 \\ x_1 + 2x_2 &= 4 \\ -3x_1 &= -4 && \Rightarrow \\ x_1 &= \frac{4}{3} && \Rightarrow \end{aligned}$$

$$\begin{aligned} x_2 &= \frac{4}{3} && \Rightarrow \\ D &= \left(\frac{4}{3}, \frac{4}{3}, 0, 0\right) && \text{feasible} \\ z &= -\frac{16}{3} \end{aligned}$$

$$\begin{aligned} \text{BV} &= \{x_2, s_1\} \\ \text{NBV} &= \{x_1, s_2\} \\ x_2 + s_1 &= 4 \\ 2x_2 &= 4 \\ x_2 &= 2 && \Rightarrow \\ s_1 &= 2 && \Rightarrow \\ D &= (0, 2, 2, 0) && \text{feasible} \\ z &= -6 \end{aligned}$$

(c) Draw the feasible region of problem 1 in the (x_1, x_2)-plane. Where are the basic solutions from part (b)? Which feasible solutions satisfy $s_1 = 0$? Which feasible solutions satisfy $s_2 = 0$?

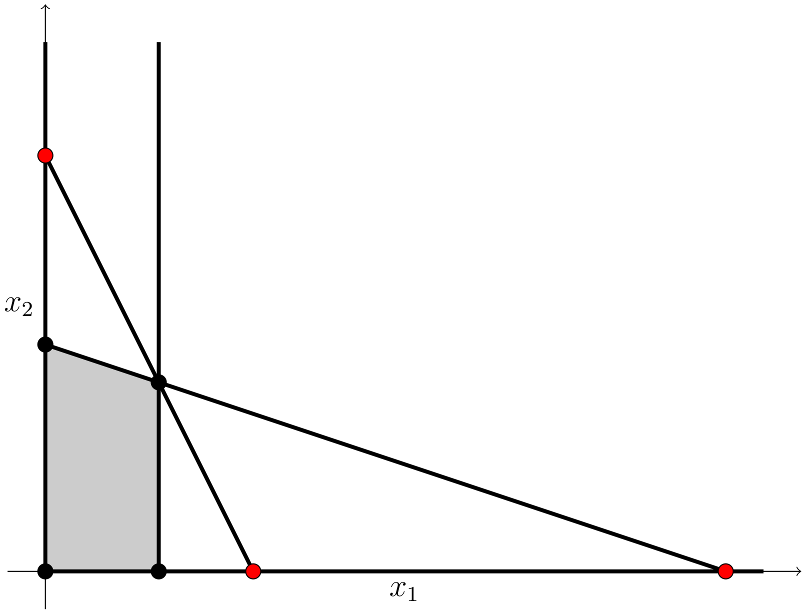


3. Consider the basic solution from exercise 2 (b) that has x_1 and x_2 as basic variables.
   (a) Determine the basic representation for this basic solution.
   (b) Is this basic solution optimal? Justify your answer both graphically (see exercise 2 (c)) and from the basic representation.
   (c) Find a non-basic variable such that increasing its value improves the objective value. How much can we increase the value of this basic variable without leaving the feasible region? Which is the resulting basic solution? Is this solution optimal?

Lecture 5

Degenerate Basic Solutions

Consider a variation of our first example, where the $x \leq 4$ constraint has changed to $x \leq 3$. One property we can quickly see is that there are now three lines intersecting a specific point $(3, 5)$, compared to the two of standard points. Essentially, the coordinates of this point are determined by more constraints than strictly necessary, and the three index sets that would typically identify different points now identify the same point. This can cause some of the basic variables to also be set to zero, in this case $x_3 = x_4 = x_5 = 0$.



We can define a basic solution as **degenerate** if one or more basic variables are zero. This means that a degenerate basic solution has more than $n - m$ zero-valued variables, therefore if we look at the tableau, there exists at least a basic variable such that $i \in I$ and $y_{i,0} = 0$.

On the other hand, we can define a basic solution as **non-degenerate** if all of its basic variables are different from zero.

Finite Termination Theorem

The theorem states that if all basic feasible solutions are **non-degenerate**, then the simplex algorithm must terminate after a **finite** number of steps, either with an optimal solution, or a proof that the problem is unbounded. The proof is as follows;

- Due to non-degeneracy, we have $\forall i \in I \; [y_{i,0} > 0]$
- Unless we find an optimal solution, or detect unboundedness in steps 1 or 2, we have;

  $$\beta_0' = \beta_0 - \frac{\beta_q}{y_{p,q}} y_{p_0} < \beta_0$$

  Looking at the signs, we know that $\beta_q > 0$ (from how we choose the variable x_q entering the basis). Similarly, we know that $y_{p,q} > 0$, otherwise we have an unbounded LP (since we choose the minimum value, and if all the bounds are ∞, we will encounter the unbounded case). Finally, we also know that $y_{p,0}$ must also be positive, hence the entire product is strictly positive.
- As such, we have a strictly decreasing sequence of objective values;

  $$\beta_0 > \beta_0' > \beta_0'' > \dots$$

  This results in no basic solution being repeated.
- Since there are $\binom{n}{m}$ ways to pick m columns out of n to form an index set I, we can say that there are $\leq \binom{n}{m}$ basic solutions
- As such, the process cannot continue indefinitely and must terminate at either step 1 or 2 after a finite number of iterations

Degeneracy

We have the following lemma; assuming that $\forall i \in [1, n]$, there exists a basic solution $\hat{x}$, with $\hat{x}_i \neq 0$. Then a basic solution x is **degenerate** iff it is associated with **more than one index set**. The proof is as follows (proving that if the basic solution x has more than one index set $\Rightarrow$ the basic solution x is degenerate);

- Suppose x corresponds to the index sets I_1 and I_2, where $I_1 \neq I_2$
- Then $x_i = 0$ for all nonbasic variables x_i, and $i \notin I_1$, $i \notin I_2$ (or both)
- Since $I_1 \neq I_2$, there must be a nonbasic variable x_i in I_1, which is a basic variable in I_2, and since the two sets describe the same basic solution x, x_i must be 0 in I_2 where it is basic (note that this also holds the other way around)
- Therefore, x is a degenerate basic solution

For example, consider the basic solution x corresponding to index sets $I_1 = \{1, 2, 3\}$ and $I_2 = \{1, 2, 4\}$. Therefore $x_4 = 0$ due to I_1, and $x_3 = 0$ due to I_2.

The steps to prove the other direction of the implication (where a degenerate basic solution $x \Rightarrow$ the basic solution has more than one index set) is as follows;

- Consider the corresponding simplex tableau, where x is a degenerate basic solution corresponding to the index set I
- Due to degeneracy, we have $\exists p \in I \; [y_{p,0} = 0]$
- There must also $\exists q \notin I \; [y_{p,q} \neq 0]$, otherwise we would always have $x_p = 0$ in all the feasible set (impossible with the theorem statement)

  Consider the following table;

BS	$\cdots$	x_p	$\cdots$	x_q	$\cdots$	RHS
x_p	$\cdots$	1	$\cdots$	$y_{p,q}$	$\cdots$	0

  Note that if $y_{p,q} = 0$ (such that the only non-zero element in the row is $y_{p,p} = 1$), we would have $x_p = 0$.
- Since we know this exists, we can pivot on (p, q), which gives a new basic solution;

  $$y'_{q,0} = \frac{y_{p,0}}{y_{p,q}} = 0 = y_{p,0} \text{ and also } \forall i \in I \setminus \{p\} \left[y'_{i,0} = y_{i,0} - \frac{y_{i,q}}{y_{p,q}} y_{p,0} = y_{i,0} \right]$$

  This basic solution is identical to the current one
- Therefore, x corresponds to both the index set I and $(I \setminus \{p\}) \cup \{q\}$

This breaks the finite termination of the simplex algorithm since;

- The index sets I and $(I \setminus \{p\}) \cup \{q\}$ produce the same basic feasible solution, but with different basic representations
- If we pivot on (p, q), when $y_{p,0} = 0$, the new basic feasible solution is identical to the old one
- Therefore, we have no strict monotonic improvement of the objective value since;

  $$\beta'_0 = \beta_0 - \frac{\beta_q}{y_{p,q}} y_{p,0} = \beta_0 - \frac{\beta_q}{y_{p,q}} 0 = \beta_0$$
- We denote a pivot step (p, q) as **degenerate** if $y_{p,0} = 0$, and **non-degenerate** otherwise

We can then decompose the simplex algorithm into a sequence of degenerate pivots, followed by a non-degenerate pivot, followed by a sequence of degenerate pivots. Note that some / all of these degenerate pivot sequences can be empty. Geometrically, the basic feasible solution remains unchanged through a sequence of degenerate pivots, and only moves to a different BFS on a non-degenerate pivot.

Cycling

We know that there is a finite number of index sets if **no index set is repeated**, as the number of index sets is $\leq \binom{n}{m}$. However, pivoting can result in cycling behaviour in some rare instances. Generally, choosing degenerate pivots is a necessary condition, but not a sufficient one, for cycling. After a sequence of pivots we return to the same index set, causing cycling.

Consider the following example;

$$\begin{aligned} \text{minimize} \qquad & z = -\frac{3}{4}x_4 + 20x_5 - \frac{1}{2}x_6 + 6x_7 \\ \text{subject to} \qquad & x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0 \\ & x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = 0 \\ & x_3 + x_6 = 1 \end{aligned}$$

When $I = \{1, 2, 3\}$, we have the following tableau;

BV	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
z	0	0	0	$\frac{3}{4}$	-20	$\frac{1}{2}$	-6	0
x_1	1	0	0	$\frac{1}{4}$	-8	-1	9	0
x_2	0	1	0	$\frac{1}{2}$	-12	$-\frac{1}{2}$	3	0
x_3	0	0	1	0	0	1	0	1
z	-3	0	0	0	4	$\frac{7}{2}$	-33	0
x_4	4	0	0	1	-32	-4	36	0
x_2	-2	1	0	0	4	$\frac{3}{2}$	-15	0
x_3	0	0	1	0	0	1	0	1
z	-1	-1	0	0	0	2	-18	0
x_4	-12	8	0	1	0	8	-84	0
x_5	$-\frac{1}{2}$	$\frac{1}{4}$	0	0	1	$\frac{3}{8}$	$-\frac{15}{4}$	0
x_3	0	0	1	0	0	1	0	1
z	2	-3	0	$-\frac{1}{4}$	0	0	3	0
x_4	$-\frac{3}{2}$	1	0	$\frac{1}{8}$	0	1	$-\frac{21}{2}$	0
x_2	$\frac{1}{16}$	$-\frac{1}{8}$	0	$\frac{3}{64}$	1	0	$\frac{3}{16}$	0
x_3	$\frac{3}{2}$	-1	1	$\frac{1}{8}$	0	0	$\frac{21}{2}$	1
				$\vdots$				
z	0	0	0	$\frac{3}{4}$	-20	$\frac{1}{2}$	-6	0
x_1	1	0	0	$\frac{1}{4}$	-8	-1	9	0
x_2	0	1	0	$\frac{1}{2}$	-12	$-\frac{1}{2}$	3	0
x_3	0	0	1	0	0	1	0	1

We can avoid cycling by amending the pivoting convention. Bland's rule states the following (with this rule, the algorithm cannot cycle and is therefore finite);

(i) choose the **lowest-numbered** (leftmost) nonbasic column q with a positive cost (instead of choosing the largest β);

$$q = \min\{j \neq 0 \mid \beta_j > 0\}$$

(ii) Denote as p the row with minimal $\bar{x}_{i,q}$, choosing the smallest index in the case of a tie (same as standard convention)

Degeneracy in Practice

More recent experience with larger problems indicates that cycling occurs (while still being a rare event). Remedies such as Bland's rule are not satisfactory as it increases the number of iterations (and therefore time) in problems where cycles do not occur. In practice, it's possible to introduce a small perturbation by **replacing** a $y_{i,0} = 0$ with $y_{i,0} = \epsilon > 0$, and continue from there.