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;

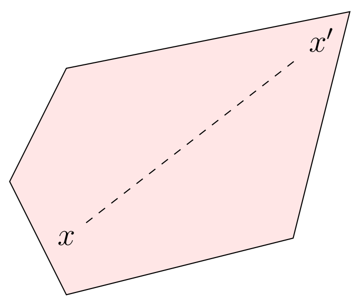
$$\min_{\boldsymbol{x}} z = f(\boldsymbol{x}) \qquad \text{subject to } \boldsymbol{x} \in \mathcal{X}$$

- **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

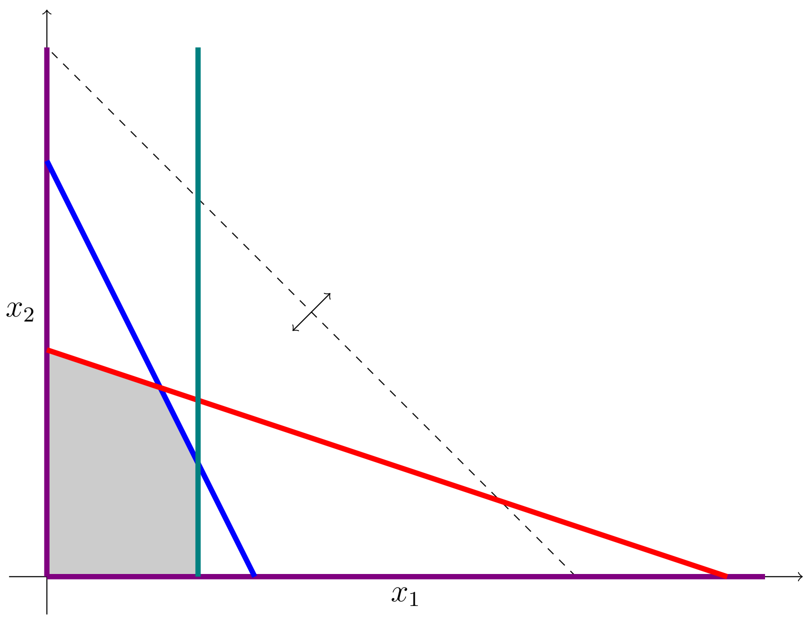
- C sells for £1000 per ton
- a maximum of 4 tons of A can be sold per week

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;

$$
\begin{aligned}
O &= (0, 0) \\
P &= (0, 6) \\
Q &= (3, 5) \\
R &= (4, 3) \\
S &= (4, 0)
\end{aligned}
$$

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;

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

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;

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

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 $\boldsymbol{Ax} \leq \boldsymbol{b}$ and $\boldsymbol{x} \geq 0$ is equivalent to the same minimisation subject to $\boldsymbol{Ax} + \boldsymbol{s} = \boldsymbol{b}$ and $\boldsymbol{x}, \boldsymbol{s} \geq 0$. The slack variables take the value of the difference $\boldsymbol{b} - \boldsymbol{Ax}$. 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 $\boldsymbol{Ax} - \boldsymbol{b}$. 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 $I = \{1, \dots, m\}$.

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

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;

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

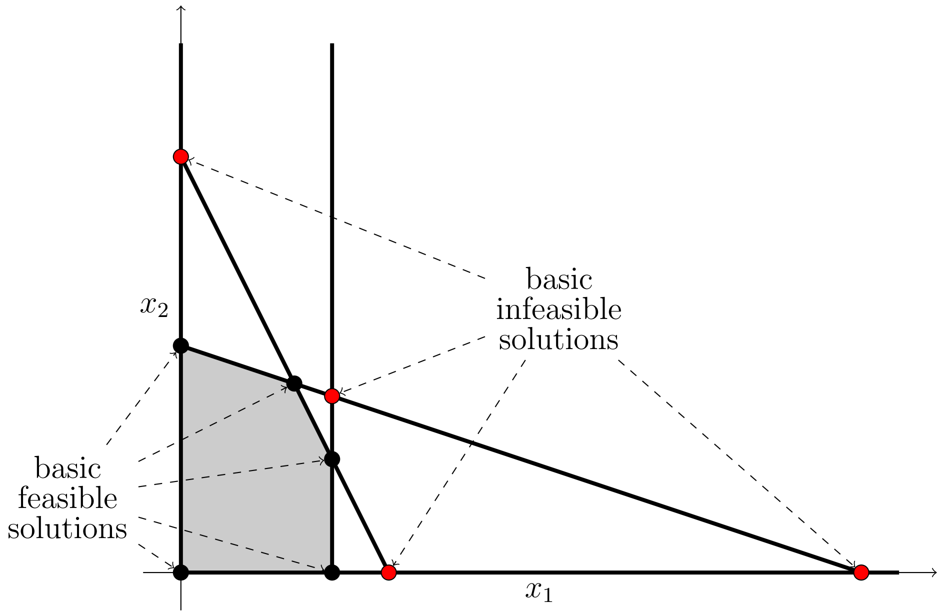
This $\boldsymbol{x}$ is the **unique basic solution** to $\boldsymbol{Ax} = \boldsymbol{b}$ with respect to I. However - this doesn't mean it's feasible, as we could end up with negative values. The geometric intuition that the corners of the feasible set correspond are LP come back into play, when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

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

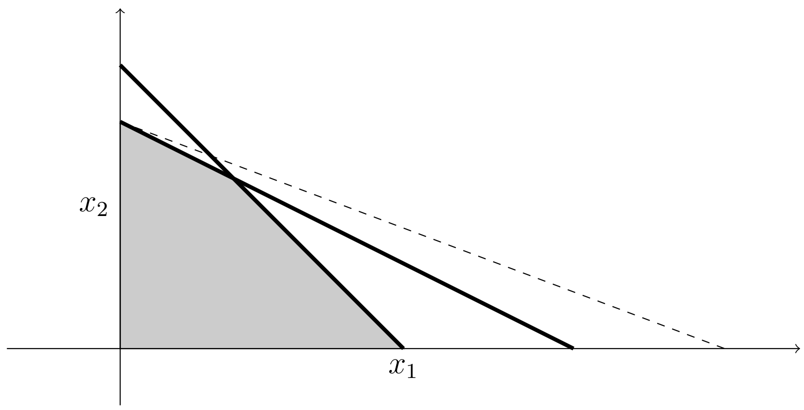
This gives an obvious, but very inefficient, method through a finite search. The number of distinct BFS is usually less than that upper bound however, as $B(I)$ may be singular (non-invertible), or the corresponding BS may not be feasible.

Example

Consider the following optimisation problem;

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

This has the following graphical representation;



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

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

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 $\boldsymbol{A} = [\boldsymbol{a_1}, \ldots, \boldsymbol{a_n}]$. where $\boldsymbol{a_i} \in \mathbb{R}^m$ is the i^{th} column of $\boldsymbol{A}$. Take any index set $I \subseteq \{1, \ldots, m\}$ 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	$y_{p,1}$	$\cdots$	$y_{p,p}$	$\cdots$	$y_{p,q}$	$\cdots$	$y_{p,n}$	$y_{p,0}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\vdots$

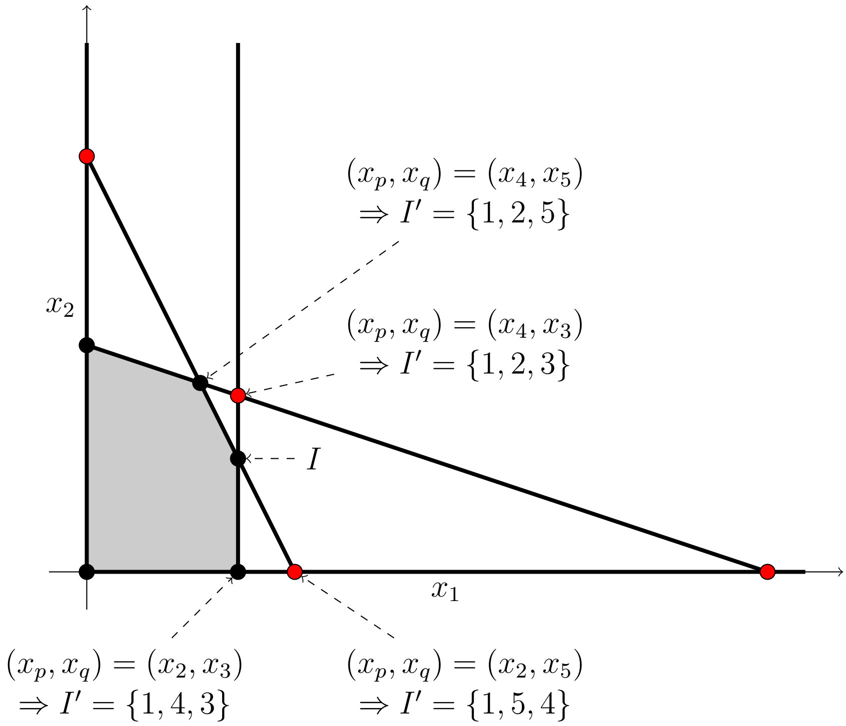
This has the following properties;

- $\forall i \in I\ y_{i,i} = 1$, 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 $I = \{1, 2, 4\}$;



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;

0. find initial BFS and its basic representation
1. if $\beta_i \leq 0$ for all $i \notin I$; we can stop, the current BFS is optimal
2. 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
3. choose x_q with the largest $\beta_q > 0$ (x_q enters the basis)
4. choose $p \in \text{argmin}_{i \in I} \bar{x}_{i,q}$ (x_p leaves the basis)
5. pivot on $y_{p,q}$ and repeat from step 1

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.

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

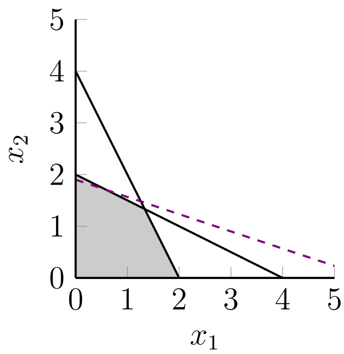
(b) For the problem in standard form, determine all basic solutions. Which of these problems are feasible, and what are their objective values?

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

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

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

(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.

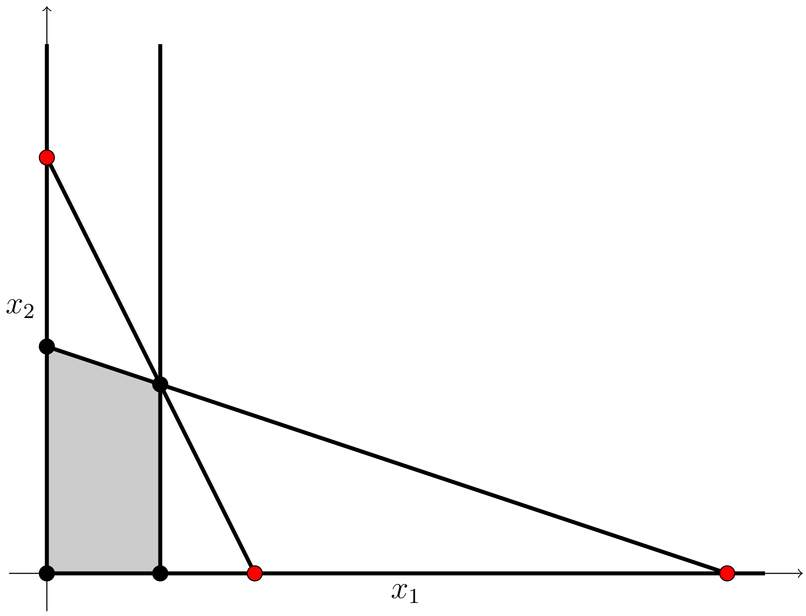
   (b) Is this basic solution optimal? Justify your answer both graphically (see exercise 2 (c)) and from the basic representation.

   (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;

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

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

$\vdots$

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

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.

Lecture 6

Initial Basic Feasible Solution

In the initial step (step 0) of the simplex algorithm, we require an **initial** BFS and the corresponding basic representation.

We can consider the "all slack basis";

$$\begin{aligned}
\text{minimize} \quad & z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{subject to} \quad & a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n + x_{n+1} = b_1 \\
& a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n + x_{n+2} = b_2 \\
& \vdots \\
& a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n + x_{n+m} = b_m \\
& x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m} \geq 0
\end{aligned}$$

Therefore we can take a basic representation for $I = n+1, \dots, n+m$, which is feasible of $\forall i \in [1, m]$ $b_i \geq 0$. However, this is not always the case.

If we now consider an example without an obvious initial BFS, a system with equalities and inequalities in both direction, assuming all variables and RHS's are non-negative;

$$\begin{aligned}
x_1 + x_2 + x_3 &= 10 \\
2x_1 - x_2 &\geq 2 \\
x_1 - 2x_2 + x_3 &\leq 6 \\
x_1, x_2, x_3 &\geq 0
\end{aligned}$$

We can now standardise it by adding slack variables and subtracting surplus variables. Note that here we denote slack variables in red and surplus variables in blue;

$$\begin{aligned}
x_1 + x_2 + x_3 &= 10 \\
2x_1 - x_2 - x_4 &= 2 \\
x_1 - 2x_2 + x_3 + x_5 &\leq 6 \\
x_1, x_2, x_3, x_4, x_5 &\geq 0
\end{aligned}$$

Here we have no basic feasible representation, since only slack variables behave like basic variables. Another approach to the above is to introduce artificial variables to the original equalities and $\geq$ inequalities;

$$\begin{aligned}
x_1 + x_2 + x_3 + \xi_1 &= 10 \\
2x_1 - x_2 - x_4 + \xi_2 &= 2 \\
x_1 - 2x_2 + x_3 + x_5 &\leq 6 \\
x_1, x_2, x_3, x_4, x_5, \xi_1, \xi_2 &\geq 0
\end{aligned}$$

The artificial variables behave like basic variables, and therefore we have a basic feasible representation. However, this system is not equivalent to the original one - when the artificial variables are zero the set of solutions is the same, however when they are strictly positive we will have more solutions. If we can find a basic feasible solution such that $\xi_1, \xi_2 = 0$, then we have found a basic feasible solution for the original LP.

To find such a solution, we solve the **auxiliary LP**;

$$\begin{aligned}
\text{minimize} \quad & \zeta = \xi_1 + \xi_2 \\
\text{subject to} \quad & x_1 + x_2 + x_3 + \xi_1 = 10
\end{aligned}$$

$$
\begin{aligned}
2x_1 - x_2 - x_4 + \xi_2 &= 2 \\
x_1 - x_2 + x_3 + x_5 &= 6 \\
x_1, x_2, x_3, x_4, x_5, \xi_1, \xi_2 &\geq 0
\end{aligned}
$$

Clearly, the minimum value we can achieve for ζ is 0. The initial BFS for this LP is given by $\xi_1 = 10, \xi_2 = 2, x_5 = 6$. If we are able to minimise ζ to 0, we have a basic feasible solution for the original LP, and if not, we cannot satisfy the original LP (it is infeasible).

However, we need a basic representation for the initial BFS. The objective function $\zeta = \xi_1 + \xi_2$ is expressed in terms of the basic variables. To express ζ as a function of the nonbasic variables we add all equations with artificial variables to the objective;

$$
\begin{aligned}
\zeta - \xi_1 - \xi_2 &= 0 & (+) \\
x_1 + x_2 + x_3 + \xi_1 &= 10 & (+) \\
2x_1 - x_2 - x_4 + \xi_2 &= 2 & (=) \\
\zeta + 3x_1 + x_3 - x_4 &= 12 &
\end{aligned}
$$

This auxiliary LP is feasible and bounded by construction, therefore the algorithm must terminate in step 1 with an optimal BFS, in one of two cases;

- $\zeta = 0$ - this implies $\xi_1 = \xi_2 = 0$, and the optimal BFS of the auxiliary LP is a BFS for the original LP
- $\zeta > 0$ - the auxiliary LP has no feasible solution with $\xi_1 = \xi_2 = 0$, hence the original system has no BFS, therefore it is infeasible

We can solve the auxiliary LP as follows;

BV	x_1	x_2	x_3	x_4	x_5	ξ_1	ξ_2	RHS
ζ	3	0	1	-1	0	0	0	12
ξ_1	1	1	1	0	0	1	0	10
ξ_2	2	-1	0	-1	0	0	1	2
x_5	1	-2	1	0	1	0	0	6
ζ	0	$\frac{3}{2}$	1	$\frac{1}{2}$	0	0	$-\frac{3}{2}$	9
ξ_1	0	$\frac{3}{2}$	1	$\frac{1}{2}$	0	1	$-\frac{1}{2}$	9
x_1	1	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0	0	$\frac{1}{2}$	1
x_5	0	$-\frac{3}{2}$	1	$\frac{1}{2}$	1	0	$-\frac{1}{2}$	5
ζ	0	0	0	0	0	-1	-1	0
x_2	0	1	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$-\frac{1}{3}$	6
x_1	1	0	$\frac{1}{3}$	$-\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	4
x_5	0	0	2	1	1	1	-1	14

Since we've now found an optimal solution, and confirmed that $\zeta = 0$ at this point, we have $I = \{2, 1, 5\}$ as a BFS for the original system. We can also take the basic representation and omit ζ, ξ_1, ξ_2 for phase 2.

Two Phase Simplex Algorithm

The first phase is as follows;

1. modify the constraints so that all RHS's are non-negative (multiply by -1 if a constraint has a negative RHS)
2. identify all equality and $\geq$ constraints
3. standardise inequalities (add slacks for $\leq$, subtract excesses for $\geq$)
4. add artificial constraints ξ_i to the constraints identified in step 2

5. let ζ be the sum of all artificial variables and derive the basic representation for ζ
6. find the minimum of ζ using the simplex algorithm

There are three cases for the second phase, the trivial of which is when $\zeta^* > 0$, from which we can state that the original LP is infeasible. However, for the non-trivial case (when all ξ_i are nonbasic at optimality), we perform the following steps;

1. remove all artificial columns from the optimal phase 1 tableau
2. derive the basic representation for z (original objective) with respect to the optimal index set from phase 1
3. solve the original LP with the simplex algorithms, using the final basis of phase 1 as the initial basis of phase 2 - the optimal of phase 2 is the optimal of the original LP

On the other hand, when there is at least one basic ξ_i at optimality;

1. as $\zeta^* = 0$, we can conclude all $\xi_i = 0$, and therefore we have some basic variables equal to zero
2. we have found a degenerate BFS for the original LP, and a basic representation for the auxiliary problem
3. since the BFS is degenerate, we can pivot on $y_{p,q} \neq 0$, corresponding to an artificial ξ_p and original variable x_q, and keep $\zeta^* = 0$
4. all ξ_i variables can therefore be removed from the basis, obtaining a BFS for the original LP

Lecture 7

Example

We can modify the running example as follows;

$$\begin{aligned}
\text{maximize} \quad & y = 3x_1 + 4x_2 \\
\text{subject to} \quad & x_1 + x_2 \leq 4 \\
& 2x_1 + x_2 \leq 5 \\
& x_2 \geq 1 \\
& x_1, x_2 \geq 0
\end{aligned}$$

We can first transform this into the standard form as follows (by adding slack variables and subtracting an excess variable), and then adding artificial variables;

$$\begin{aligned}
-\text{ minimize} \quad & z = -3x_1 - 4x_2 \\
\text{subject to} \quad & x_1 + x_2 + x_3 = 4 \\
& 2x_1 + x_2 + x_4 = 5 \\
& x_2 - x_5 + x_6 = 1 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}$$

This forms the auxiliary LP, where we can take the initial index set $I = \{3, 4, 6\}$;

$$\begin{aligned}
\text{minimize} \quad & \zeta = x_6 \\
\text{subject to} \quad & x_1 + x_2 + x_3 = 4 \\
& 2x_1 + x_2 + x_4 = 5 \\
& x_2 - x_5 + x_6 = 1 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}$$

We can now form the following simplex tableau (for phase 1);

BV	ζ	x_1	x_2	x_3	x_4	x_5	x_6	RHS
ζ	1	0	1	0	0	-1	0	1
x_3	0	1	1	1	0	0	0	4
x_4	0	2	1	0	1	0	0	5
x_6	0	0	1	0	0	-1	1	1
ζ	1	0	0	0	0	0	-1	0
x_3	0	1	0	1	0	1	-1	3
x_4	0	2	0	0	1	1	-1	4
x_2	0	0	1	0	0	-1	1	1

From this, we have $\zeta^* = 0$, with an index set of $I = \{3, 4, 2\}$ - also proving that the original LP is feasible. Note that we can also use the part of the phase 1 tableau in teal for our second phase.

Min-Max Problems

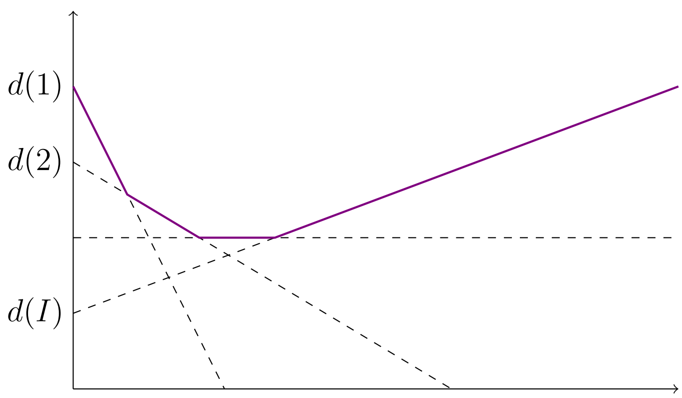
Consider a family of linear functions: $y_i(x) = c(i)^\top x + d(i)$, and also set

$$\phi(x) = \max_{i=1,\dots,I} \left\{ c(i)^\top x + d(i) \right\} \text{ for } c(i) \in \mathbb{R}^n, d(i) \in \mathbb{R}$$

Then the following is called a **min-max problem**;

$$\begin{aligned} \text{minimize} \quad & \phi(x) \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Consider the following scalar case, where $n = 1$ (note that the lines have corresponding $c(i)$ gradients), and $\phi(x)$ is the line in violet;



However, we can convert a Min-Max (MM) problem into a linear program (LP) as follows;

$$\begin{aligned} \text{minimize} \quad & z \\ \text{subject to} \quad & z \geq c(i)^\top x + d(i) \quad \forall i = 1, \dots, I \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

If (x_{LP}^*, z_{LP}^*) is an optimal solution of this LP, then x_{LP}^* is also an optimal solution of Min-Max, and Min-Max has optimal value $\phi(x_{LP}^*) = z_{LP}^*$.

We can first check that an optimal solution in LP, (x_{LP}^*, z_{LP}^*), is also feasible in MM. It satisfies the constraint $Ax = b$, since it is also present in LP, and similarly will also satisfy $x \geq 0$, therefore it is feasible. From here, we can form a proof by contradiction, that there exists a better solution in MM than the one obtained from x_{LP}^*;

$$\exists x_{MM}^* \; [\phi_{MM}^* = \phi(x_{MM}^*) < \phi(x_{LP}^*)]$$

We can validate that this x^*_{MM} also obviously satisfies the $Ax = b$ and $x \geq 0$ constraints in LP, since it is also present in MM. However - we need to ensure that it satisfies the additional constraint in LP;

$$\phi^*_{MM} = \max_{j=1,\dots,I} \left\{ c(j)^\top x^*_{MM} + d(j) \right\}$$

Since we are taking the maximal value, by definition of ϕ, we can be certain it also satisfies that constraint. Going back to our result for z^*_{LP}, we have the following;

$$z^*_{LP} \geq \max_{i=1,\dots,I} \left\{ c(i)^\top x^*_{LP} + d(i) \right\} = \phi(x^*_{LP})$$

By our assumption, we have a better value in $\phi(x^*_{MM})$, therefore;

$$z^*_{LP} \geq \phi(x^*_{LP}) > \phi(x^*_{MM})$$

This however causes a contradiction, since we have also proven that x^*_{MM} is also feasible in LP, hence it would be the optimal value, such that $z^*_{LP} = \phi(x^*_{MM})$.

Tutorial

1. It's a long worded question (hence not typing it out).

   Note that the profits are divided by a thousand, and the hours are divided by 10 for brevity.

$$\begin{aligned} \text{maximize} \quad & y = 6x_1 + 4x_2 + 3x_3 \\ \text{subject to} \quad & 4x_1 + 5x_2 + 3x_3 \leq 12 \\ & 3x_1 + 4x_2 + 2x_3 \leq 10 \\ & 4x_1 + 2x_2 + x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

   This results in the following, after conversion to standard form;

$$\begin{aligned} -\ \text{minimize} \quad & z = -6x_1 - 4x_2 - 3x_3 \\ \text{subject to} \quad & 4x_1 + 5x_2 + 3x_3 + x_4 = 12 \\ & 3x_1 + 4x_2 + 2x_3 + x_5 = 10 \\ & 4x_1 + 2x_2 + x_3 + x_6 = 8 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{aligned}$$

   We can now use an all slack basis, hence $I = \{4, 5, 6\}$;

BV	x_1	x_2	x_3	x_4	x_5	x_6	RHS	ratios
z	6	4	3	0	0	0	0	
x_4	4	5	3	1	0	0	12	3
x_5	3	4	2	0	1	0	10	$\frac{10}{3}$
x_6	4	2	1	0	0	1	8	2
z	0	1	$\frac{3}{2}$	0	0	$-\frac{3}{2}$	-12	
x_4	0	3	2	1	0	-1	4	2
x_5	0	$\frac{5}{2}$	$\frac{5}{4}$	0	1	$-\frac{3}{4}$	4	$\frac{16}{5}$
x_1	1	$\frac{1}{2}$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	2	8

$$\vdots$$

$$\begin{aligned} \boldsymbol{x}^* &= \left(\frac{3}{2}, 0, 2, 0, \frac{3}{2}, 0\right) \\ z^* &= -15 \\ y^* &= £15,000 \end{aligned}$$

Lecture 8

Example of Min-Max

$$
\begin{aligned}
\text{minimize} \quad & \phi(\boldsymbol{x}) \\
\text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\
& \boldsymbol{x} \geq 0
\end{aligned}
$$

Where we have the following;

$$
\begin{aligned}
\phi(\boldsymbol{x}) &= \max_{i=1,\dots,I} \{\boldsymbol{c}(\boldsymbol{i})^\top \boldsymbol{x}\} \\
\boldsymbol{c}(\boldsymbol{i}) &\in \mathbb{R}^m \\
\boldsymbol{A} &\in \mathbb{R}^{m\times m} \\
\boldsymbol{b} &\in \mathbb{R}^m
\end{aligned}
$$

in the specific case;

$$
\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} 10 & 5 \\ 5 & 9 \end{bmatrix} \\
\boldsymbol{b} &= \begin{bmatrix} -2 \\ 5 \end{bmatrix} \\
\boldsymbol{c}(\boldsymbol{i}) &= \begin{bmatrix} 5-i \\ 3-i \end{bmatrix}
\end{aligned}
$$

Converted to the LP equivalent, we have the following;

$$
\begin{aligned}
\text{minimize} \quad & z \\
\text{subject to} \quad & z \geq 4x_1 + 2x_2 \\
& z \geq 3x_1 + x_2 \\
& 10x_1 + 5x_2 = -2 \\
& 5x_1 + 9x_2 = 5 \\
& x1, x2 \geq 0
\end{aligned}
$$

Min-Min Problems

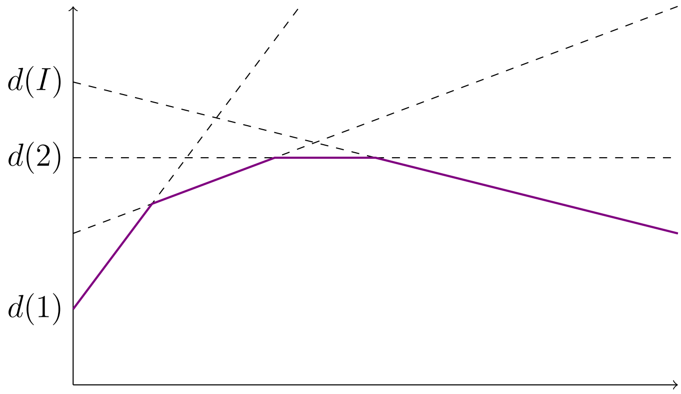
Similarly, we can set the following (with a family of linear equations);

$$
\psi(x) = \min_{i=1,\dots,I} \left\{c(i)^\top x + d(i)\right\} \text{ for } c(i) \in \mathbb{R}^n, d(i) \in \mathbb{R}
$$

Then the following is called a **min-min problem**;

$$
\begin{aligned}
\text{minimize} \quad & \psi(x) \\
\text{subject to} \quad & Ax = b \\
& x \geq 0
\end{aligned}
$$

Consider the following scalar case, where $n = 1$ (note that the lines have corresponding $c(i)$ gradients), and $\psi(x)$ is the line in violet;



Consider a set of I linear programs, $LP(1), \dots, LP(I)$, with $LP(i)$ defined as;

$$
\begin{aligned}
\text{minimize} \quad & z_i = c(i)^\top x(i) + d(i) \\
\text{subject to} \quad & Ax(i) = b \\
& x(i) \geq 0
\end{aligned}
$$

Denote z_i^* as the optimal solution for $LP(i)$, and let $LP(j)$ be the LP that has the minimal objective;

$$z_j^* = \min_{i=1,\dots,I} z_i^*$$

Let $x^*(j)$ be its optimal solution, then it is optimal in MM (min-min), and we have the following result;

$$\psi^* = \psi(x^*(j)) = z_j^*$$

Interchangeability of Min-Operations

We observe the following lemma; let X and Y be arbitrary sets, and let $f : X \times Y \to \mathbb{R}$ and arbitrary function defined on $X \times Y$, then we have;

$$\min_{x \in X} \min_{y \in Y} f(x, y) = \min_{y \in Y} \min_{x \in X} f(x, y)$$

The lemma then implies that by finding the minimum of each of the individual linear equations, the minimum of which is the same as finding the minimum of ψ.

Fractional Linear Programming

Consider the following fractional linear program (FLP);

$$\min \left\{ \frac{\alpha_0 + \alpha_1 x_1 + \dots + \alpha_n x_n}{\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n} \,\middle|\, \boldsymbol{Ax} = \boldsymbol{b}; \boldsymbol{x} \geq 0 \right\}$$

We make the following assumptions;

- the feasible set of the FLP is bounded;

$$\forall \boldsymbol{x} : \boldsymbol{Ax} = \boldsymbol{b}; \boldsymbol{x} \geq 0 \ \ [\exists L > 0 \ \ [\ ||x|| \leq L\ ]]$$

- the denominator of the objective function is strictly positive (the following holds for all feasible $\boldsymbol{x}$)

$$\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n > 0$$

Homogenisation

Let us introduce new variables $y_i \geq 0$ for $i = 1, \dots, n$ and $y_0 > 0$. Adding the new variable y_0 gives us an additional degree of freedom. If we set $x_i = \frac{y_i}{y_0}$, we can **homogenise** the FLP as follows;

$$\begin{aligned} \text{minimize} \quad & \frac{\alpha_0 y_0 + \alpha_1 y_1 + \cdots + \alpha_n y_n}{\beta_0 y_0 + \beta_1 y_1 + \cdots + \beta_n y_n} \\ \text{subject to} \quad & b_i y_0 - \sum_{j=1}^{n} a_{i,j} y_j = 0 \quad \forall i = 1, \dots, m \\ & y_0 > 0 \\ & y_1, \dots, y_n \geq 0 \end{aligned}$$

For any $(y_0, y_1, \dots, y_n)$ feasible in HFLP, $\lambda(y_0, y_1, \dots, y_n)$ is also feasible (with $\lambda > 0$) - and it will have the same objective value. For each $(y_0, \dots, y_n)$, we can find a λ that satisfies the following;

$$\beta_0 y_0 + \cdots + \beta_n y_n = 1$$

The scaled point will have identical objective, thus we can restrict our attention to those points and find an optimal solution, hence the denominator in the objective of HFLP can always be **normalised** to unity. The **normalised problem** is as follows;

$$\begin{aligned} \text{minimize} \quad & \alpha_0 y_0 + \alpha_1 y_1 + \cdots + \alpha_n y_n \\ \text{subject to} \quad & \beta_0 y_0 + \sum_{j=1}^{n} \beta_j y_j = 1 \\ & b_i y_0 - \sum_{j=1}^{n} a_{i,j} y_j = 0 \quad \forall i = 1, \dots, m \\ & y_0 > 0 \\ & y_1, \dots, y_n \geq 0 \end{aligned}$$

The first constraint forces the denominator of the objective of HFLP to be equal to 1.

We denote the optimal solution of the normalised problem as; $(y_0^*, y_1^*, \dots, y_n^*)$. The optimal solution for FLP is (construction only works if $y_0^* \neq 0$);

$$\left(\frac{y_1^*}{y_0^*}, \dots, \frac{y_n^*}{y_0^*} \right)$$

There is also a derivation for relaxing the lower bound on y_0, however we do not need to know it.

Example of FLP

$$\begin{aligned} \text{minimize} \quad & \frac{x_1 + 2x_2}{4x_1 + 3x_2 + 3} \\ \text{subject to} \quad & x_1 + x_2 \leq 2 \\ & -x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We can first argue that it is certainly bounded, due to the first constraint (upper bound), and a lower bound from the non-negativity constraint. Converting to HFLP, we use the following;

$$x_1 = \frac{y_1}{y_0} \text{ and } x_2 = \frac{y_2}{y_0}$$

$$
\begin{aligned}
\text{minimize} \quad & \frac{y_1 + 2y_2}{4y_1 + 3y_2 + 3y_0} \\
\text{subject to} \quad & y_1 + y_2 \leq 2y_0 \\
& -y_1 + y_2 \leq 1y_0 \\
& y_1, y_2 \geq 0 \\
& y_0 > 0
\end{aligned}
$$

Converting, and setting the denominator to 1, we get the following;

$$
\begin{aligned}
\text{minimize} \quad & y_1 + 2y_2 \\
\text{subject to} \quad & 4y_1 + 3y_2 + 3y_0 = 1 \\
& y_1 + y_2 + s_1 = 2y_0 \\
& -y_1 + y_2 + s_2 = 1y_0 \\
& y_1, y_2, s_1, s_2 \geq 0 \\
& y_0 > 0
\end{aligned}
$$

Finally, we also need to state the following;

$$
x^* = \left(\frac{y_1^*}{y_0^*}, \frac{y_2^*}{y_0^*}\right)
$$

GLPK

This looks at Klee-Minty cubes. There is a technique called steepest edge that calculates the angle between the edge and the objective function. However this is more computationally expensive, and since we've noticed that there are points in which we might not move in the simplex algorithm, there is preference for moving faster than being precise.

Case study 4 is also mentioned, regarding a dataset concerning Italian wines.

Tutorial

1. In linear programming show that a variable that becomes nonbasic in one iteration of the simplex method cannot become basic in the next iteration. Consider the sign of the coefficient in the objective row for the variable that becomes nonbasic.

   We have x_i being basic at iteration k, and x_i being nonbasic at iteration $k+1$. The variable that became basic (after pivoting) is x_j at iteration $k+1$, with a pivot of (i, j). At iteration k, we have $\beta_i^{(k)} = 0$, since it was a basic variable, hence has zeroes in the entire column other than its corresponding row. Similarly, we also know that $y_{i,j}^{(k)} > 0$ by the simplex algorithm. Looking at iteration $k+1$;

$$
\begin{aligned}
\beta_i^{(k+1)} &= \beta_i^{(k)} - \frac{y_{i,i}^{(k)}}{y_{i,j}^{(k)}} \beta_j^{(k)} \\
&= 0 - \frac{1}{y_{i,j}^{(k)} \beta_j^{(k)}} \\
&< 0
\end{aligned}
$$

Lecture 9

Introduction to Duality

For every optimisation problem we encounter, duality allows us to construct another optimisation problem. The first problem is the primal problem, and the constructed one the dual problem.

Dual Problem

This starts with a flow diagram, but that will take too long to draw out. The first optimisation problem is to work out the maximum flow from a source node to a termination node, and in this example the answer was 11 (from identifying bottlenecks). The dual problem was to minimum cost to disrupt all flow from the source to the destination (the cost is equal to the flow of a path) - which was also 11.

Another motivation is the following problem;

$$\begin{aligned}
\text{maximize} \quad & z = x_1 + 6x_2 && \text{(1a)}\\
\text{subject to} \quad & x_1 \leq 200 && \text{(1b)}\\
& x_2 \leq 300 && \text{(1c)}\\
& x_1 + x_2 \leq 400 && \text{(1d)}\\
& x_1, x_2 \geq 0 && \text{(1e)}
\end{aligned}$$

As a sanity check, by adding 1 of equation 1b and 6 of equation 1c, we have $x_1 + x_2 \leq 2000$. However, if we wanted to bring the upper bound further, we can take some linear combination of equation 1b, 1c, and 1d. In this example, the best combination is to take 5 times 1c and 6 times 1d, which gives us the lowest upper bound of 1900.

If we were to introduce a multiplier for each constraint, (y_1, y_2, y_3) for equations 1b, 1c, and 1d respectively, we can do the following. Note that $y_1, y_2, y_3 \geq 0$ to preserve inequalities.

By multiplying the constraints, and summing them, we obtain the following;

$$x_1 + 6x_2 \leq (y_1 + y_3)x_1 + (y_2 + y_3)x_3 \leq 200y_1 + 300y_2 + 400y_3$$

This also needs to form an upper bound on our objective function;

$$\begin{aligned}
y_1 + y_3 &\geq 1\\
y_2 + y_3 &\geq 6
\end{aligned}$$

From this, we have created the dual LP;

$$\begin{aligned}
\text{minimize} \quad & 200y_1 + 300y_2 + 400y_3\\
\text{subject to} \quad & y_1 + y_3 \geq 1\\
& y_2 + y_3 \geq 6\\
& y_1, y_2, y_3 \geq 0
\end{aligned}$$

It's important to observe the following results;

$$\begin{aligned}
(x_1, x_2) &= (100, 300) && \text{primal}\\
\text{primal optimal} &= 1900\\
(y_1, y_2, y_3) &= (0, 5, 1) && \text{dual}\\
\text{dual optimal} &= 1900
\end{aligned}$$

Note that the number of constraints in the primal becomes the number of variables in the dual, and similarly the number of variables in the primal becomes the number of constraints in the dual.

We have the following symmetric definition (note that the dual of (D) is (P));

- **primal problem** (P) $\qquad \boldsymbol{c}, \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq 0\}$$

- **dual problem** (D) $\qquad \boldsymbol{c}, \boldsymbol{A}, \boldsymbol{b}$ same as (P), $\boldsymbol{y} \in \mathbb{R}^m$

$$\min\{\boldsymbol{b}^\top \boldsymbol{y} : \boldsymbol{A}^\top \boldsymbol{y} \geq \boldsymbol{c}, \boldsymbol{y} \geq 0\}$$

Weak Duality

Weak duality holds for any programming, whether it is linear or non-linear. Assume that both the problems (P) and (D), as described above, are both feasible. Let $\boldsymbol{x} \in \mathbb{R}^n$ be feasible for (P), and $\boldsymbol{y} \in \mathbb{R}^m$ be feasible for (D), then we have the result that;

$$\boldsymbol{c}^\top \boldsymbol{x} \leq \boldsymbol{b}^\top \boldsymbol{y}$$

The above inequality holds for **any feasible solution**, not just the optimal solutions.

The proof is as follows;

- (P) requires that $\boldsymbol{Ax} \leq \boldsymbol{b} \Rightarrow \boldsymbol{y}^\top \boldsymbol{Ax} \leq \boldsymbol{y}^\top \boldsymbol{b}$ (since $\boldsymbol{y} \geq 0$)
- similarly (D) implies that $\left(\boldsymbol{A}^\top \boldsymbol{y}\right)^\top \geq \boldsymbol{c}^\top \Rightarrow \boldsymbol{y}^\top \boldsymbol{Ax} \geq \boldsymbol{c}^\top \boldsymbol{x}$ (since $\boldsymbol{x} \geq 0$)

Then if both are feasible, we obtain the following;

$$\boldsymbol{c}^\top \boldsymbol{x} \leq \boldsymbol{y}^\top \boldsymbol{Ax} \leq \underbrace{\boldsymbol{y}^\top \boldsymbol{b} = \boldsymbol{b}^\top \boldsymbol{y}}_{\text{becomes a scalar}}$$

This states that the value of the primal is always upper bounded by the value of the dual.

Strong Duality

For linear programs (generally convex), under a few assumptions, the optimal values for the dual and primal are equal.

Assume the problem (P) is feasible with a bounded optimum. Let B be the optimal basis for (P), with an optimal basic solution (x_B^*, x_N^*), we then have the following;

- $\boldsymbol{y}^* = \left(\boldsymbol{B}^{-1}\right)^\top \boldsymbol{c_B}$ is an **optimal** solution for (D)
- $\boldsymbol{c}^\top \boldsymbol{x}^* = \boldsymbol{b}^\top \boldsymbol{y}^*$ - the objective values coincide

Note that if (P) is unbounded, then (D) is infeasible (and vice versa).

Note that we also refer to the optimal solution of the dual problem, $\boldsymbol{y}^* = \left(\boldsymbol{B}^{-1}\right)^\top \boldsymbol{c_B}$, as the shadow price of the primal problem;

$$\boldsymbol{\Pi} = \boldsymbol{y}^* = \left(\boldsymbol{B}^{-1}\right)^\top \boldsymbol{c_B}$$

Consider the following forms of the primal and dual;

Primal (P):

$$\begin{aligned} \text{maximize} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{Ax} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Dual (D):

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{b}^\top \boldsymbol{y} \\ \text{subject to} \quad & \boldsymbol{A}^\top \boldsymbol{y} \geq \boldsymbol{c} \\ & \boldsymbol{y} \geq 0 \end{aligned}$$

Consider the following possibilities with a primal / dual pair;

		primal		
		finite optimal	unbounded	infeasible
dual	finite optimal	yes, possible	no, by strong duality	no, by strong duality
	unbounded	no, by strong duality	no, by weak duality (1)	
	infeasible	no, by strong duality		

(1) imagine the drawing of the line; for both to be unbounded, they will cross over - dual goes to $-\infty$, whereas primal goes to ∞

Lecture 10

Indirect Way

The approach here is to take an original problem, convert it to the primal problem we've seen before, convert it to the dual;

$$(P') \rightarrow (P) \rightarrow (D) \rightarrow (D')$$

For this, we will be working with the following example (where we attempt to find the dual);

$$\begin{aligned} \text{maximize} \quad & 2x_1 + x_2 \\ \text{subject to} \quad & x_1 + x_2 = 2 \\ & 2x_1 - x_2 \geq 3 \\ & x_1 - x_2 \leq 1 \\ & x_1 \geq 0 \\ & x_2 \in \mathbb{R} \end{aligned}$$

The algorithm is as follows;

1. bring LP to either (P) or (D)
   - replace any unbounded variables (not positive) $x_i \in \mathbb{R}$ with $(x_i^+ - x_i^-)$, where $x_i^+, x_i^- \geq 0$

$$\begin{aligned} \text{maximize} \quad & 2x_1 + x_2^+ - x_2^- \\ \text{subject to} \quad & x_1 + x_2^+ - x_2^- = 2 \\ & 2x_1 - x_2^+ + x_2^- \geq 3 \\ & x_1 - x_2^+ + x_2^- \leq 1 \\ & x_1, x_2^+, x_2^- \geq 0 \end{aligned}$$

   - replace any equality constraints with two inequality constraints

$$\begin{aligned} \text{maximize} \quad & 2x_1 + x_2^+ - x_2^- \\ \text{subject to} \quad & x_1 + x_2^+ - x_2^- \leq 2 \\ & x_1 + x_2^+ - x_2^- \geq 2 \\ & 2x_1 - x_2^+ + x_2^- \geq 3 \\ & x_1 - x_2^+ + x_2^- \leq 1 \\ & x_1, x_2^+, x_2^- \geq 0 \end{aligned}$$

   - change any constraint directions with negation if necessary

$$\begin{aligned} \text{maximize} \quad & 2x_1 + x_2^+ - x_2^- \\ \text{subject to} \quad & x_1 + x_2^+ - x_2^- \leq 2 \\ & -x_1 - x_2^+ + x_2^- \leq -2 \\ & -2x_1 + x_2^+ - x_2^- \leq -3 \\ & x_1 - x_2^+ + x_2^- \leq 1 \\ & x_1, x_2^+, x_2^- \geq 0 \end{aligned}$$

   - change direction of objective function with negation if necessary
2. obtain dual according to definition

- if LP is now in form (P), dual is (D)

  With this, we have the following primal problem;

$$\boldsymbol{c} = \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2^+ \\ x_2^- \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & -1 \\ -1 & -1 & 1 \\ -2 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} 2 \\ -2 \\ -3 \\ 1 \end{bmatrix}$$

  by the definition, we have the the following;

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\boldsymbol{A}^\top = \begin{bmatrix} 1 & -1 & -2 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

  Therefore, our dual problem is;

$$\begin{aligned} \text{minimize} \quad & 2y_1 - 2y_2 - 3y_3 + y_4 \\ \text{subject to} \quad & y_1 - y_2 - 2y_3 + y_4 \geq 2 \\ & y_1 - y_2 + y_3 - y_4 \geq 1 \\ & -y_1 + y_2 - y_3 + y_4 \geq -1 \\ & y_1, y_2, y_3, y_4 \geq 0 \end{aligned}$$

- vice versa; if LP is in form (D), dual is (P)

3. simplify dual problem (optional)

   - replace variable pairs $y_i, y_j \geq 0, i \neq j$ that occur in **all** functions as $\alpha y_i - \alpha y_j$ with $y_k \in \mathbb{R}$
   - replace matching inequality constraints with a single equality constraint

   Note that in the example above, we notice that y_1, y_2 always show up in pairs, and therefore we can replace it with $y_1' \in \mathbb{R}$, where $y_1' = y_1 - y_2$. Similarly, after negation of the third constraint, we have a matching inequality pair, thus we end up with the following simplified problem;

$$\begin{aligned} \text{minimize} \quad & 2y_1' - 3y_3 + y_4 \\ \text{subject to} \quad & y_1' - 2y_3 + y_4 \geq 2 \\ & y_1' + y_3 - y_4 = 1 \\ & y_3, y_4 \geq 0 \\ & y_1' \in \mathbb{R} \end{aligned}$$

Direct Way

Since the detour to transform to (P) and then to (D) may be tedious, there is a way to apply duality without the detour. We need to recall the rules from the transformation;

1. for every primal constraint, create one dual variable, and for every primal variable, create one dual constraint
2. the dual coefficient matrix is $\boldsymbol{A}^\top$, the former right-hand sides $\boldsymbol{b}$ now become the new costs, and the former costs $\boldsymbol{c}$ become the new new RHS's
3. if the primal is a maximisation problem, then the dual is a minimisation problem
   - if the i^{th} primal constraint is $\geq, =, \leq$, the i^{th} dual variable becomes $y_i \leq 0, y_i \in \mathbb{R}, y_i \geq 0$ respectively
   - if the j^{th} primal variable is $x_j \geq 0, x_j \in \mathbb{R}, x_j \leq 0$, the j^{th} dual constraint becomes $\geq, =, \leq$ respectively
4. if the primal is a minimisation problem, then the dual is a maximisation problem (note the different directions of the inequalities in the dual)
   - if the i^{th} primal constraint is $\geq, =, \leq$, the i^{th} dual variable becomes $y_i \geq 0, y_i \in \mathbb{R}, y_i \leq 0$ respectively
   - if the j^{th} primal variable is $x_j \geq 0, x_j \in \mathbb{R}, x_j \leq 0$, the j^{th} dual constraint becomes $\leq, =, \geq$ respectively

Sensitivity Analysis

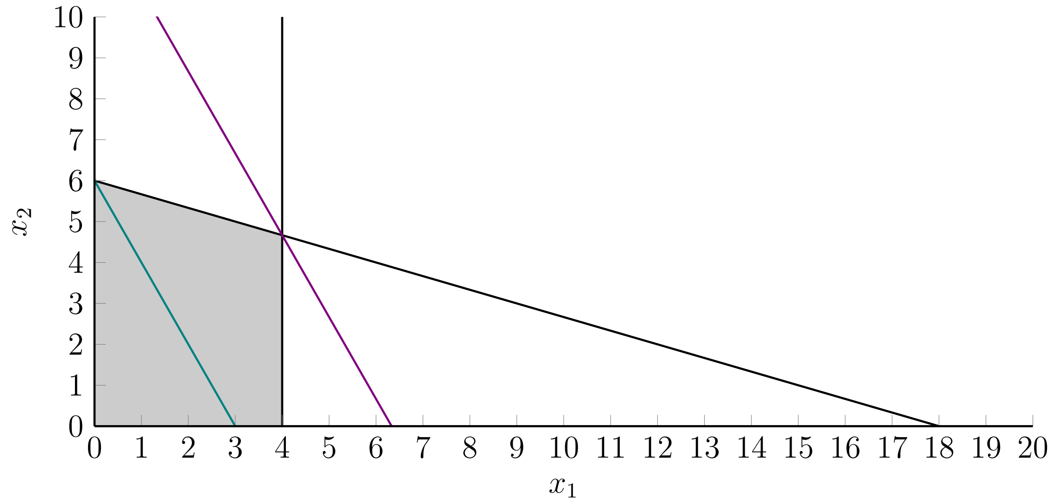
Sensitivity is about understanding how the solution of the linear program depends on certain parameters. Note that so far we've assumed that we know all the constraints, although this isn't always the case. Consider the following maximisation problem, where the constraints are the availability of X, the availability of Y, and the demand of x_1. In this case, we do not know the precise availability of X, hence we call it p_1;

$$\begin{aligned} \text{maximize} \quad & y = x_1 + x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \leq p_1 \\ & x_1 + 3x_2 \leq 18 \\ & x_1 \leq 4 \\ & x_1, x_2 \geq 0 \end{aligned}$$

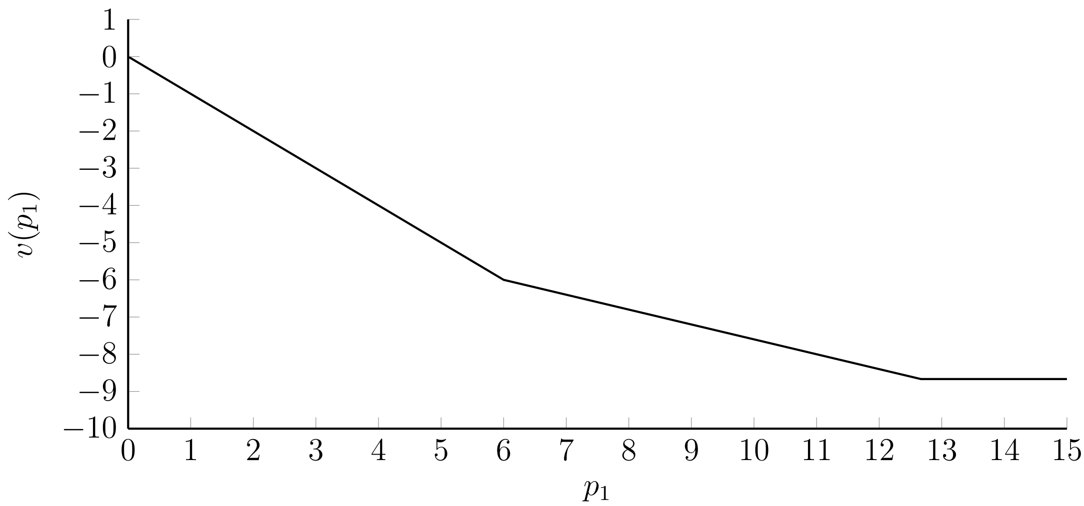
In standard form;

$$\begin{aligned} -\text{minimize} \quad & z = -x_1 - x_2 \\ \text{subject to} \quad & 2x_1 + x_2 + x_3 = p_1 \\ & x_1 + 3x_2 + x_4 = 18 \\ & x_1 + x_5 = 4 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

We can introduce the idea of a value function, $v(p_1)$, which expresses the optimal value of the LP as a function of the unknown parameter. If we graphically consider the above example;



Note the that when $p_1 < 0$, the problem is infeasible. In this case, the important points to consider are when $p_1 = \frac{38}{3}$ and $p_1 = 6$. We can also graph $v(p_1)$ as follows;



Let $\boldsymbol{p} \in \mathbb{R}^m$ denote a general RHS, and define the value function $v(\boldsymbol{p}) : \mathbb{R}^m \to \mathbb{R}$ as;

$$v(\boldsymbol{p}) = \min\left\{z = \boldsymbol{c}^\top \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{p}; \boldsymbol{x} \geq 0\right\}$$

Therefore, if we were to solve the original LP (**reference problem**), we would compute $v(\boldsymbol{b})$.

Suppose we have solved the reference problem, where $\boldsymbol{p} = \boldsymbol{b}$, and found an optimal basis which satisfies the following;

- **feasibility** $\quad \boldsymbol{x_B} = \boldsymbol{B}^{-1}\boldsymbol{b} \geq 0$
- **optimality** $\quad \boldsymbol{r} = \boldsymbol{c_N} - \boldsymbol{N}^\top \left(\boldsymbol{B}^{-1}\right)^\top \boldsymbol{c_B} \geq 0$

Suppose we are given $\boldsymbol{p} \in \mathbb{R}^m$. If $\boldsymbol{B}^{-1}\boldsymbol{p} \geq 0$, we therefore know that $\boldsymbol{B}$ is also feasible for $\boldsymbol{p}$; we also know that it is **optimal**, since the reduced cost vector ($\boldsymbol{r}$) does not depend on $\boldsymbol{b}$. This gives us $x_B(\boldsymbol{p}) = \boldsymbol{B}^{-1}\boldsymbol{p}$.

Recall the vector of shadow prices, $\boldsymbol{\Pi} \in \mathbb{R}^m$, defined as;

$$\boldsymbol{\Pi} = \left(\boldsymbol{B}^{-1}\right)^\top \boldsymbol{c_B}$$

Note that since there can be more than one optimal basis, the shadow prices **do not need to be unique**. The shadow prices give us information about the sensitivity of the value function $v(\boldsymbol{p})$ at $\boldsymbol{p} = \boldsymbol{b}$.

Behaviour of Value Function

The theorem states the following, for all $\boldsymbol{p} \in \mathbb{R}^m$ with $\boldsymbol{B}^{-1}\boldsymbol{p} \geq 0$;

$$v(\boldsymbol{p}) = v(\boldsymbol{b}) + \boldsymbol{\Pi}^\top(\boldsymbol{p} - \boldsymbol{b})$$

We can prove this by first stating that if $\boldsymbol{B}^{-1}\boldsymbol{p} \geq 0$, then $\boldsymbol{B}$ remains as the optimal basis since $\boldsymbol{r}$ is not affected by the change from $\boldsymbol{b}$ to $\boldsymbol{p}$ (see above). Therefore we find the following;

$$\begin{aligned} v(\boldsymbol{p}) &= \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{p} \\ &= \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{b} + \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}(\boldsymbol{p} - \boldsymbol{b}) \\ &= v(\boldsymbol{b}) - \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}(\boldsymbol{p} - \boldsymbol{b}) \end{aligned}$$

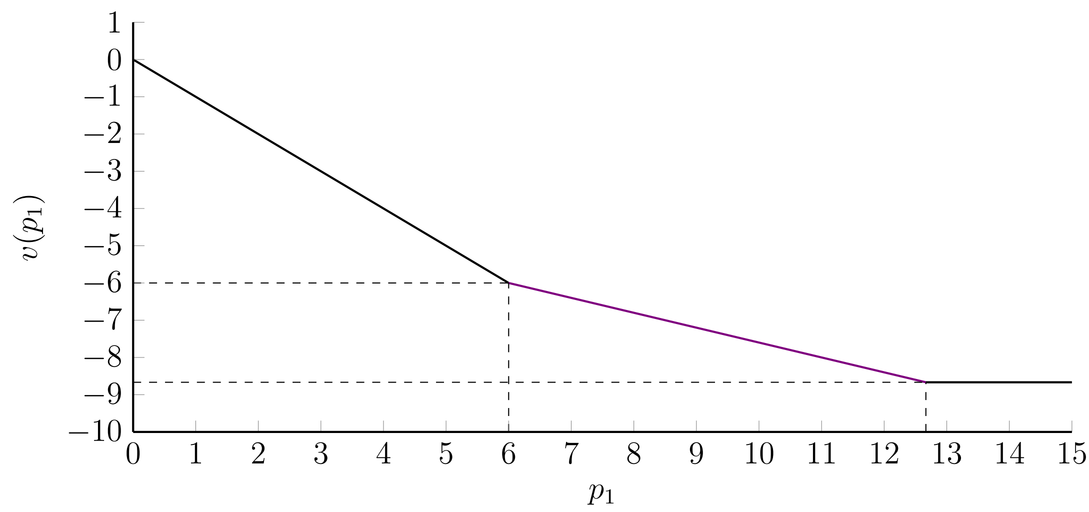
More generally, we have the following theorem, for all $\boldsymbol{p} \in \mathbb{R}^m$;

$$v(\boldsymbol{p}) \geq v(\boldsymbol{b}) + \boldsymbol{\Pi}^\top(\boldsymbol{p} - \boldsymbol{b})$$

The proof is as follows;

$$\begin{aligned} v(\boldsymbol{p}) &= \min\left\{\boldsymbol{c}^\top\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{p}; \boldsymbol{x} \geq 0\right\} \\ &= \min\left\{\boldsymbol{c}^\top\boldsymbol{x} - \boldsymbol{\Pi}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{p}) \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{p}; \boldsymbol{x} \geq 0\right\} && \text{note that } \boldsymbol{A}\boldsymbol{x} - \boldsymbol{p} = \boldsymbol{0} \\ &\geq \min\left\{\boldsymbol{c}^\top\boldsymbol{x} - \boldsymbol{\Pi}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{p}) \mid \boldsymbol{x} \geq 0\right\} && \text{relaxed constraint on feasible set} \\ &= \min\left\{\left(\boldsymbol{c}^\top - \boldsymbol{\Pi}^\top\boldsymbol{A}\right)\boldsymbol{x} + \boldsymbol{\Pi}^\top\boldsymbol{p} \mid \boldsymbol{x} \geq 0\right\} \\ &= \boldsymbol{\Pi}^\top\boldsymbol{p} + \underbrace{\min\left\{\left(\boldsymbol{c}^\top - \boldsymbol{\Pi}^\top\boldsymbol{A}\right)\boldsymbol{x} \mid \boldsymbol{x} \geq 0\right\}}_{\text{always } \geq 0} \\ &\geq \boldsymbol{\Pi}^\top\boldsymbol{p} \\ &= \boldsymbol{\Pi}^\top\boldsymbol{b} + \boldsymbol{\Pi}^\top(\boldsymbol{p} - \boldsymbol{b}) \\ &= \boldsymbol{c_B}^\top \boldsymbol{B}^{-1}\boldsymbol{b} + \boldsymbol{\Pi}^\top(\boldsymbol{p} - \boldsymbol{b}) \\ &= v(\boldsymbol{b}) - \boldsymbol{\Pi}^\top(\boldsymbol{p} - \boldsymbol{b}) \end{aligned}$$

It's important to note that the shadow price ($\boldsymbol{\Pi}$) is computed for the specific value of $\boldsymbol{b}$. Recall the graph of $v(p_1)$;



Let $p_1 = b_1 = 11$. If the optimal basis stays the same (in the region in violet), the optimal costs change by $\Pi_1 = -\frac{2}{5}$ if the availability of X increases by 1. Therefore, we can state that the derivative of the value function at that point is equal to the shadow price, which is true in general (other than at the degenerate points).

We can apply this to the real example as follows. If the company can buy an additional amount of time on machine X, at a price of μ_1 per unit, there are two cases;

- if $\mu_1 + \Pi_1 < 0$, the overall cost **decreases**, and therefore it is worthwhile
- if $\mu_1 + \Pi_1 > 0$, the overall cost **increases**, and therefore it is not worthwhile

As such, we can conclude that $-\Pi_1$ is the maximum price one should pay for a unit of time on machine X. Similarly, the maximum amount of money a worker should ask from a company is equal to the shadow price, as otherwise it wouldn't make sense.

We can also write the new constraints' RHS as;

$$\boldsymbol{p} = \boldsymbol{b} + \xi \boldsymbol{e_t}$$

Where $\boldsymbol{e_t}$ is a vector with all 0, except for a single 1 at the t^{th} row / position. Accepting the offer causes the total production cost to become;

$$v(\boldsymbol{b}) + \mu_t \xi \begin{cases} = v(\boldsymbol{b}) + \Pi_t \xi & \text{if } \boldsymbol{B}^{-1}(\boldsymbol{b} + \xi e_t) \geq 0 \\ \geq v(\boldsymbol{b}) + \Pi_t \xi & \text{in general} \end{cases}$$

For the equality case, we can accept the offer if $\mu_t + \Pi_t < 0$, and reject it otherwise. We cannot threshold this simply in the case where we encounter a different optimal basis.

Evaluation of Shadow Prices

We have the following lemma. Suppose that row t is initially a $\leq$-constraint, and a slack variable x_s had been added. Then we have $\Pi_t = \beta_s$, where β_s is the objective coefficient of x_s in the optimal tableau.

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

If x_s is nonbasic in the final tableau, we have;

$$\begin{aligned} \beta_s &= -r_s \\ &= -\boldsymbol{c_N}^\top \boldsymbol{e_s} + \boldsymbol{c_B}^\top \left(\boldsymbol{B}^{-1}\right) \boldsymbol{N} \boldsymbol{e_s} \\ &= -c_s + \boldsymbol{\Pi}^\top \boldsymbol{e_t} \\ &= 0 + \boldsymbol{\Pi}^\top \boldsymbol{e_t} \\ &= \Pi_t \end{aligned}$$

Note that we are able to obtain $\boldsymbol{N}\boldsymbol{e_s} = \boldsymbol{e_t}$. The multiplication by $\boldsymbol{e_s}$ gives us the s^{th} column, however by construction (since it is a slack variable), we will get $\boldsymbol{e_t}$.

On the other hand, if x_s is basic, we take the following steps;

$$\begin{aligned} \beta_s &= 0 \\ &= c_s \\ &= \boldsymbol{e_s}^\top \boldsymbol{c_B} \\ &= \boldsymbol{e_s}^\top \boldsymbol{B}^\top \boldsymbol{\Pi} && \boldsymbol{\Pi} = \left(\boldsymbol{B}^{-1}\right)^\top \boldsymbol{c_B} \Rightarrow \boldsymbol{c_B} = \boldsymbol{B}^\top \boldsymbol{\Pi} \\ &= \boldsymbol{e_t}^\top \boldsymbol{\Pi} \\ &= \Pi_t \end{aligned}$$

Another lemma we have is as follows. Suppose we row t is a $\geq$-constraint, and a surplus x_s has been added. Then $\Pi_t = -\beta_s$, where β_s is the objective coefficient of x_s in the final tableau.

Lecture 11

This starts with the continuation of the proof started in the previous lecture.

Game Theory

We can think of game theory as an extension of single agent optimisation (the optimisation we've looked at), to a problem with multiple decision-makers. Often times, when agents act in their own best interest, the overall performance of the system is very poor.

In a "game", every player takes their own decision and the payoff each player receives depends on the choice of all players. Mathematically, we can formalise it as;

- each player $i = 1, \dots, n$ has a set of actions $x_i \in \mathcal{X}_i$ (in a single agent optimisation problem, we have $n = 1$)
- player i receives a payoff $J_i(x_1, \dots, x_i, \dots, x_n)$

A special class of this is a two-person zero-sum game, with finite actions;

- two players, a row player (RP), and a column player (CP)
- RP can choose one out of m strategies (row strategies)
- CP can choose one out of n strategies (column strategies)
- zero-sum means that if RP wins, then CP looses (and vice versa)

Payoff Matrix

We can describe a two-player zero-sum game as a payoff matrix. If RP was to play strategy i, and CP was to play strategy j, then CP pays $a_{i,j}$ to RP;

		CP			
		strategy 1	strategy 2	$\cdots$	strategy n
RP	strategy 1	$a_{1,1}$	$a_{1,2}$	$\cdots$	$a_{1,n}$
	strategy 2	$a_{2,1}$	$a_{2,2}$	$\cdots$	$a_{2,n}$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	strategy m	$a_{m,1}$	$a_{m,2}$	$\cdots$	$a_{m,n}$

For rock-paper-scissors, we can create the payoff matrix as follows (if a player loses, they pay 1 unit to the other);

		CP		
		R	P	S
RP	R	0	-1	1
	P	1	0	-1
	S	-1	1	0

Two-Person Zero-Sum Games

We make the following assumptions about these games;

1. each player knows the game setting (available strategies and the values of the payoff matrix)
2. both players simultaneously choose their strategy, without knowing what their opponent chooses
3. each players chooses a strategy that enables them to do their best, reasoning as if the opponent could anticipate their strategy
4. both players are rational; they try to maximise their utility and show no compassion for the opponent

Consider this running example, where we have the following election;

- two players; RP and CP
- both players have three strategies;

- L - campaign the last two days in London
  - B - campaign the last two days in Birmingham
  - S - split the last two days, campaign one day in London and the other in Birmingham
- the payoff is how many voters does RP acquire from CP

		CP		
		L	B	S
	L	1	2	4
RP	B	1	0	5
	S	0	1	-1

We observe that L will always be better than S for RP, and therefore we can conclude that RP will never play strategy S. Since both players will realise this, we can ignore it.

		CP		
		L	B	S
RP	L	1	2	4
	B	1	0	5

Similarly, since CP is attempting to minimise, we know that CP will never play S (since L is always better) thus we can eliminate it with the same reasoning as above.

		CP	
		L	B
RP	L	1	2
	B	1	0

Similar to the first observation, RP will never play B, and thus it can be eliminated. After that row is eliminated, we can see that CP will never play B either (since it will be 1 versus 2, and CP aims to minimise). This allows us to conclude, with **dominant strategy equilibrium** that both players will go for strategy L.

Dominance

We can formalise our definition of dominance as follows;

- **dominated row strategy**

  Row strategy i is dominated by row strategy i' if $a_{i',j} \geq a_{i,j}$ for all column strategies $j = 1, \dots, n$, and $a_{i',j} > a_{i,j}$ for at least one j
- **dominated column strategy**

  Column strategy j is dominated by column strategy j' if $a_{i,j'} \leq a_{i,j}$ for all row strategies $i = 1, \dots, m$, and $a_{i,j'} < a_{i,j}$ for at least one i
- **dominant strategy equilibrium**

  If repeated removal of dominated strategies leads to a game where each player only has one strategy left, this strategy pair is a dominated strategy equilibrium

  - if one exists, it is unique
  - if one exists, then rational players will play the associated equilibrium strategies

A rational player will never play a dominated strategy.

Nash Equilibrium

We assume that each player chooses a strategy that enables them to do the best in face of a worst-case opponent (security strategy). We define α_i as the payoff of row strategy i, when facing the worst case opponent. The worst-case is when the column player is minimising the gain of the row player;

$$\alpha_i = \min_{j=1,\dots,n} a_{i,j}$$

Therefore, the row player should choose the strategy i that maximises the worst-case payoff;

$$\max_{i=1,\dots,m} \min_{j=1,\dots,n} a_{i,j}$$

Similarly, we can perform the same reasoning for the column player, where β_j is the cost of column strategy j;

$$\beta_j = \max_{i=1,\dots,m} a_{i,j}$$

And the minimised worst-case payoff is (security strategy over columns);

$$\min_{j=1,\dots,n} \max_{i=1,\dots,m} a_{i,j}$$

		CP			
		L	B	S	α_i
	L	-3	-2	6	-3
RP	B	2	0	2	0
	S	5	-2	-4	-4
	β_j	5	0	6	

The rational outcome is for both players to play (B, B). This strategy pair is called a **pure Nash equilibrium**, the corresponding payoff is referred to as the **value of the game**.

Formally, we can define a **Nash Equilibrium** as a strategy pair $(i^\star, j^\star)$ such that no player has an incentive to unilaterally (only one changes, with no coordination) deviate from their chosen strategy if told the strategy of the other player. A Nash equilibrium may not always exist in pure strategies. If $(i^\star, j^\star)$ is a Nash equilibrium, then $\alpha_{i^\star} = \beta_{j^\star}$. The payoff of the strategy pair $\alpha_{i^\star} = \beta_{j^\star}$ is called the value of the game.

Tutorial

2. Write the dual formulation of the following (primal) linear program;

$$\begin{aligned} \text{minimize} \quad & -x_1 + x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \leq 10 \\ & 3x_1 + 7x_2 \geq 20 \\ & x_1, x_2 \geq 0 \end{aligned}$$

By changing the sign of the first constraint, we obtain the following;

$$\boldsymbol{b} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{A}^\top = \begin{bmatrix} -2 & -1 \\ 3 & 7 \end{bmatrix}$$

$$\boldsymbol{c} = \begin{bmatrix} -10 \\ 20 \end{bmatrix}$$

This allows us to obtain the dual problem as follows;

$$\begin{aligned} \text{maximize} \quad & -10y_1 + 20y_2 \\ \text{subject to} \quad & -2y_1 + 3y_2 \leq -1 \\ & -y_1 + 7y_2 \leq 1 \\ & y_1, y_2 \geq 0 \end{aligned}$$

Lecture 12

Mixed Strategies

Note that this example has no Nash equilibrium in a pure strategy. Consider the odds-and-evens game, with the following simple payoff matrix;

		CP 1	CP 2
RP	1	-1	1
RP	2	1	-1

To argue that it has no pure Nash equilibrium, we can choose a point, and from there we will be able to find an improvement for one of the players, and this will cycle.

If we allow players to randomly pick strategies with equal probabilities, we have each strategy pair being played with probability 0.25. The expected value of the game is 0 for both players. There is no reason to unilaterally change probabilities, and is an example of Nash equilibrium in mixed strategies.

We can formally define this as follows;

- In a mixed strategy, we have $(p_1, \dots, p_m; q_1, \dots, q_n)$;
  - RP plays strategy i with probability $p_i \geq 0$ $\quad \sum_{i=1}^{m} p_i = 1$
  - CP plays strategy j with probability $q_j \geq 0$ $\quad \sum_{j=1}^{n} q_j = 1$
- If $p_k = 1$ or $q_k = 1$, then k is a pure strategy
- We can define the payoff of the mixed strategy $(\boldsymbol{p}, \boldsymbol{q})$ as;

$$V(\boldsymbol{p}, \boldsymbol{q}) = \underbrace{\sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_j a_{i,j}}_{\text{expected payoff / cost}}$$

- RP seeks probabilities that maximise payoff $(p_1^*, \dots, p_m^*)$
- CP seekers probabilities that minimise payoff $(q_1^*, \dots, q_n^*)$

We define a **mixed Nash equilibrium** as a pair of mixed strategies $(\boldsymbol{p}^*, \boldsymbol{q}^*)$ such that for all other mixed strategies $(\boldsymbol{p}, \boldsymbol{q})$;

$$V(\boldsymbol{p}, \boldsymbol{q}^*) \leq V(\boldsymbol{p}^*, \boldsymbol{q}^*) \leq V(\boldsymbol{p}^*, \boldsymbol{q})$$

Neither agent has any incentive in unilaterally deviating, as RP cannot increase payoff by changing $\boldsymbol{p}$, and similarly CP cannot reduce cost by changing $\boldsymbol{q}$. This is also called a saddle-point equilibrium. Note that a pure Nash equilibrium is a **subset** of a mixed Nash equilibrium.

Column Player's Perspective

From CP's perspective, they expect RP to respond with optimal p_i for any choice of q_j, how should CP choose q_j's?

$$V_{CP} = \min_{q_1,\ldots,q_n} \overbrace{\max_{p_1,\ldots,p_m} \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j a_{i,j}}_{V(\boldsymbol{p},\boldsymbol{q})}}^{V_{CP}^{in}(\boldsymbol{q})}$$

If we only consider the inner problem, it is a function of q_j, call the problem V_{CP}^{in}. We can focus on the inner problem as follows;

$$\begin{aligned} \text{minimize} \quad & V_{CP} = V_{CP}^{in}(q_1, \ldots, q_n) \\ \text{subject to} \quad & \sum_{j=1}^{n} q_j = 1 \\ & \quad q_j \geq 0 \end{aligned}$$

For any choice of q_j's, let $\alpha_i = \sum_{j=1}^{n} q_j a_{i,j}$ be row payoffs. The inner maximisation problem then becomes the following;

$$\begin{aligned} \text{maximize} \quad & V_{CP}^{in}(q_1, \ldots, q_n) = \sum_{i=1}^{m} p_i \alpha_i \\ \text{subject to} \quad & \sum_{i=1}^{m} p_i = 1 \\ & \quad p_i \geq 0 \end{aligned}$$

However, the solution is trivial, as we can set $p_i = 1$ for the largest α_i, and $p_k = 0$ when $k \neq i$. From this, we have the inner maximisation optimal value as;

$$V_{CP}^{in}(q_1, \ldots, q_n) = \max\{\alpha_1, \ldots, \alpha_m\}$$

By expanding the definition of α_i, we conclude CP is solving a min-max problem;

$$V_{CP} = \min_{q_1,\ldots,q_n} \max \left\{ \sum_{j=1}^{n} q_j a_{1,j}, \ldots, \sum_{j=1}^{n} q_j a_{m,j} \right\}$$

Subject to the probability constraints on q_j. This simplification removes p_i from the problem entirely.

$$\begin{aligned} \text{minimize} \quad & \tau \\ \text{subject to} \quad & \quad \tau \geq \sum_{j=1}^{n} q_j a_{i,j} \quad \forall i = 1, \ldots, m \\ & \sum_{j=1}^{n} q_j = 1 \\ & \quad q_j \geq 0 \end{aligned}$$

As such, we have the optimal q_j^*'s being independent of p_i^*'s.

Row Player's Perspective

Similar reasoning applies to the choice of the row player;

$$V_{RP} = \max_{p_1,\dots,p_m} \min_{q_1,\dots,q_n} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_j a_{i,j}$$

$$\begin{aligned} \text{maximize} \quad & \tau \\ \text{subject to} \quad & \tau \leq \sum_{i=1}^{m} p_i a_{i,j} \quad \forall j = 1, \dots, n \\ & \sum_{i=1}^{m} p_i = 1 \\ & p_i \geq 0 \end{aligned}$$

Minimax Theorem

With the following payoff matrix;

		CP		
		L	B	S
	L	0	-1	2
RP	B	5	4	-3
	S	2	3	-4

This leads to the following results, for $\boldsymbol{p^*}$ and $\boldsymbol{q^*}$;

$$\begin{aligned} p_L^* &= \frac{7}{10} \\ p_B^* &= \frac{3}{10} \\ p_S^* &= 0 \\ V_{RP}^* &= \frac{1}{2} \end{aligned} \qquad \Rightarrow \qquad \begin{aligned} q_L^* &= 0 \\ q_B^* &= \frac{1}{2} \\ q_S^* &= \frac{1}{2} \\ V_{CP}^* &= \frac{1}{2} \end{aligned} \qquad \Rightarrow$$

The theorem is that for every two-person zero-sum game, the RP and CP linear programs have the same optimal value;

$$V_{RP} = \max_{p_1,\dots,p_m} \min_{q_1,\dots,q_n} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_j a_{i,j} = \min_{q_1,\dots,q_n} \max_{p_1,\dots,p_m} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_j a_{i,j} = V_{CP}$$

This can be proven by duality, and we know that the problems are feasible and bounded. As a consequence, we know that a Nash equilibrium will always exist in mixed strategies, where players expect identical payoffs and neither player has an incentive to change p_i or q_j. The statement also generalises to M players.

Integer Programming

Integer programming is a mathematical programming problem where one or more variables are integer valued;

- **binary variables** (yes / no) $\quad x_j \in \{0, 1\}$
- **integer variables** (e.g. discrete amounts) $\quad x_j \in \{0, \dots, n\}$

- mix of integer and real (MILP - mixed integer linear programming)

If we name the minimum objective values at the MILP and LP as $f(x^*_{MILP})$ and $f(x^*_{LP})$, respectively, we can say;

$$f(x^*_{LP}) \leq f(x^*_{MILP})$$

This comes from the subset constraints being lifted, and therefore in the worst case the LP value can take the MILP value.

Consider the following optimisation;

$$\begin{aligned} \text{maximize} \quad & x + y \\ \text{subject to} \quad & -2x + 2y \geq 1 \\ & -8x + 10y \leq 13 \\ & x, y \geq 0 \\ & x, y \in \mathbb{N}_0 \end{aligned}$$

In this case, the optimal integer solution is far from the optimal real solution, since it is entirely possible to construct such a problem where the feasible region intersects very few integer points.

It's important to note that it's often more difficult to solve integer linear programs than linear programs, since we don't have continuity of the loss function.

Example

A company has resources $i \in \{1, \ldots, m\}$, and resource i has a limited availability of b_i. A company can undertake projects $j \in \{1, \ldots, n\}$, and project j requires $a_{i,j}$ units of resource i and gives revenue c_j. Which projects should be undertaken?

$$\begin{aligned} \text{maximize} \quad & z = \sum_{j=1}^{n} c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{i,j} x_j \leq b_j \qquad \forall i \in \{1, \ldots, m\} \\ & x_j \in \{0, 1\} \quad \forall j \in \{1, \ldots, n\} \end{aligned}$$

Consider another example where a company has m potential distribution sites, labelled $i \in \{1, \ldots, m\}$, and building site i costs f_i. The company has n customers, labelled $j \in \{1, \ldots, n\}$, whose demands d_j need to be satisfied from one or more distribution centres. The cost to satisfy an amount $x_{i,j}$ of a customer j's demand from distribution centre i is $c_{i,j}$ if i is built. Which centres should be built, and how should the demand be satisfied, to minimise cost?

$$\begin{aligned} \text{minimize} \quad & \underbrace{\sum_{i=1}^{m} f_i y_i}_{\text{build}} + \underbrace{\sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j} x_{i,j}}_{\text{dispatch}} \\ \text{subject to} \quad & \sum_{i=1}^{m} x_{i,j} = d_j \qquad \forall j \in 1, \ldots, n \\ & x_{i,j} \leq d_j y_i \qquad \forall i \in 1, \ldots, m, j \in 1, \ldots, n \\ & x_{i,j} \geq 0 \qquad \forall i \in 1, \ldots, m, j \in 1, \ldots, n \\ & y_i \in \{0, 1\} \quad \forall i \in 1, \ldots, m \end{aligned}$$

Constraint one states that the demand must be satisfied and constraint two states that $x_{i,j}$ can only exist if the facility is built.

Combinatorial Optimisation

Combinatorial optimisation problems involving finding an optimal object from a finite set of objects - enumeration becomes intractable as the problem size grows. These problems are often reducible to a few categories;

- **knapsack problem**

  Consider n items of weight w_j, $j \in \{1, \dots, n\}$, and a knapsack of capacity W. Item j has value v_j, but not all items may fit the knapsack - we want to optimise this value.

$$\begin{aligned} \text{maximize} \quad & z = \sum_{j=1}^{n} v_j x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} w_j x_j \leq W \\ & x_j \in \{0, 1\} \end{aligned}$$

- **bin-packing problem**

  Similarly, consider n items of weight w_j, $j \in \{1, \dots, n\}$, and k bins of capacity W. Our goal is to minimise the number of bins needed to store all items.

$$\begin{aligned} \text{minimize} \quad & z = \sum_{i=1}^{k} y_i \\ \text{subject to} \quad & \sum_{j=1}^{n} w_j x_{i,j} \leq W y_i \\ & \sum_{i=1}^{k} x_{i,j} = 1 \\ & x_{i,j}, y_i \in \{0, 1\} \end{aligned}$$

  The second constraint says that each item is in exactly one bin.

- (and more)

Lecture 13

Mixed Integer Linear Programming

Note that we often omit linear, especially in the context of this course. There are several subareas of independent interest;

- **Pure Integer Linear Programming** - all variables (including slack and objective) are integer
- **Binary Linear Programming** - all variables are binary
- **Mixed Integer Binary Programming** - MILP but integer variables are binary

The MILP standard for is similar to LPs, in particular $\boldsymbol{b} \geq 0$. It's also important to note that slack and excess variables in MILPs are continuous. On the other hand, with pure IP in standard form, we have integer-valued slack and excess variables;

0. apply LP standard form transformations, except addition of slack and excess variables;

   - minimisation
   - non-negative RHS

- free variables

$$
\begin{aligned}
\text{minimize} \quad & z = -\frac{1}{3}x_1 - \frac{1}{2}x_2 \\
\text{subject to} \quad & \frac{2}{3}x_1 + \frac{1}{3}x_2 \leq \frac{4}{3} \\
& \frac{1}{2}x_1 - \frac{3}{2}x_2 \leq \frac{2}{3} \\
& x_1, x_2 \geq 0 \\
& x_1, x_2 \in \mathbb{N}_0
\end{aligned}
$$

1. scale equations of the model so all coefficients are integers

$$
\begin{aligned}
\text{minimize} \quad & z' = -2x_1 - 3x_2 \\
\text{subject to} \quad & 2x_1 + x_2 \leq 4 \\
& 3x_1 - 9x_2 \leq 4 \\
& x_1, x_2 \geq 0 \\
& x_1, x_2 \in \mathbb{N}_0
\end{aligned}
$$

   Note that in this example $z = \frac{z'}{6}$.

2. insert integer slack / excess variables

$$
\begin{aligned}
\text{minimize} \quad & z' = -2x_1 - 3x_2 \\
\text{subject to} \quad & 2x_1 + x_2 + s_3 = 4 \\
& 3x_1 - 9x_2 + s_4 = 4 \\
& x_1, x_2, s_3, s_4 \geq 0 \\
& x_1, x_2, s_3, s_4 \in \mathbb{N}_0
\end{aligned}
$$

   We can do this, as the difference between two integers is still an integer.

Logical Operations

We can model logical operations on the constraints via integer variables, consider the expression;

$$\boldsymbol{a_1}^\top \boldsymbol{x} \leq b_1 \lor \boldsymbol{a_2}^\top \boldsymbol{x} \leq b_2$$

We can express this as follows, where M is a large enough constant referred to as big-M;

$$
\begin{aligned}
\boldsymbol{a_1}^\top \boldsymbol{x} &\leq b_1 + M\delta \\
\boldsymbol{a_2}^\top \boldsymbol{x} &\leq b_2 + M(1-\delta) \\
\delta &\in \{0, 1\}
\end{aligned}
$$

This also allows us to define a feasible set which isn't connected. Consider the following; assume we want to find the minimum x that satisfies $x \in [0, 1] \lor x \in [2, 4]$. We can model it as follows;

$$
\begin{aligned}
\text{minimize} \quad & x \\
\text{subject to} \quad & x \geq 0 \\
& x \leq 4 \\
& x \leq 1 + M\delta \\
& x \geq 2 - M(1-\delta)
\end{aligned}
$$

Note that we want to choose the smallest possible big-M - one that is big enough to allow program to work, but not too large as to cause numerical instability (badly conditioned).

Another constraint that we'd often like to satisfy is to satisfy k-out-of-m;

$$\boldsymbol{a_1}^\top \boldsymbol{x} \leq b_1, \boldsymbol{a_2}^\top \boldsymbol{x} \leq b_2, \dots, \boldsymbol{a_m}^\top \boldsymbol{x} \leq b_m$$

We can then express this as follows (note when $\delta_j = 1$, that constraint does **not** play a role);

$$\begin{aligned}
\boldsymbol{a_1}^\top \boldsymbol{x} &\leq b_1 \\
\boldsymbol{a_2}^\top \boldsymbol{x} &\leq b_2 \\
&\vdots \\
\boldsymbol{a_m}^\top \boldsymbol{x} &\leq b_m \\
\sum_{j=1}^{m} \delta_j &\leq m - k && \text{at least } k \text{ zeroes} \\
\delta_j &\in \{0, 1\} && \forall j \in \{1, \dots, m\}
\end{aligned}$$

Finite-Valued Variables

Assume a variable x_j can only take a finite number of values, such that $x_j \in \{p_1, \dots, p_m\}$. We can then introduce the variables $z_{j,1}, \dots, z_{j,m} \in \{0, 1\}$, where each variable "toggles" on one of the finite values. We also need to add the constraint to ensure that only one value is taken at a time;

$$z_{j,1} + \dots + z_{j,m} = 1$$

We can then do the following replacement in the objective function and all constraints;

$$x_j = p_1 z_{j,1} + \dots + p_m z_j, m$$

Tutorial

3. Consider the following Two-Player Zero-Sum game;

		CP				
		c_1	c_2	c_3	c_4	c_5
	r_1	-4	0	5	-1	2
	r_2	4	9	-5	1	-5
RP	r_3	3	-3	0	-7	5
	r_4	7	2	6	0	5
	r_5	-7	-4	8	-5	9

1. Is there a dominant strategy equilibrium? If so, find it. If not, can we at least remove dominated strategies from the problem?
   - r_4 dominates r_3
   - r_4 dominates r_1
   - c_4 dominates c_2

		CP			
		c_1	c_3	c_4	c_5
	r_2	4	-5	1	-5
RP	r_4	7	6	0	5
	r_5	-7	8	-5	9

2. Is there a pure Nash equilibrium? If so, find it.

		CP					
		c_1	c_2	c_3	c_4	c_5	α_i
	r_1	-4	0	5	-1	2	-4
	r_2	4	9	-5	1	-5	-5
RP	r_3	3	-3	0	-7	5	-7
	r_4	7	2	6	0	5	0
	r_5	-7	-4	8	-5	9	-7
	β_j	7	9	8	1	9	

Note that this could be solved with more ease with the previous result (removed dominated). The best-worst case choice is (r_4, c_4). However, this is not a pure Nash equilibrium, as players would want to deviate from that (RP could go to r_2).

Lecture 14

The "simplest" possible ILP, feasibility with binary decisions, is NP-complete; there is no polynomial time algorithm to solve it.

Continuous Relaxation

LP relaxation is the linear program obtained by replacing all integer variables $x_j \in \mathbb{N}_0$ in an ILP with continuous variables $x_j \in \mathbb{R}$. This will have a better or same optimal value as the ILP.

The procedure is as follows;

- solve a LP relaxation (this contains all original feasible solutions, as well as others)
- if the optimal solution is integer, we are done
- otherwise, tighten the LP relaxation and repeat

  Tightening is to restrict the feasible set of the LP relaxation without excluding the optimum solution of the ILP.

Cutting Plane Algorithm

The algorithm terminates after a finite number of iterations - the resulting x^* is integer and optimal.

0. write the ILP in standard form
1. solve the LP relaxation
2. if the resulting optimal solution x^* is integer, stop as we've found the optimal
3. generate a **cut** - a constraint satisfied by all feasible integer solutions but not by previous solution x^* with non-integer components
4. add cut to the LP relaxation and go back to step 1

Consider the following problem;

$$\begin{aligned} \text{maximize} \quad & y = 5x_1 + 8x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 6 \\ & 5x_1 + 9x_2 \leq 45 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{N}_0 \end{aligned}$$

We can then add a cut $2x_1 + 3x_2 \leq 15$ - it excludes the LP optimum, without removing any feasible integer points. This is added to the LP relaxation, and we now obtain an optimal solution in $x^* = (3, 3)$ (this is actually wrong).

Gomory Cut

Assume that $x_1, \dots, x_n \geq 0$ and integer. We can define the floor function as follows;

$$\lfloor c \rfloor = \max\{a \in \mathbb{Z} : a \leq c\}$$

$$\begin{aligned}\lfloor -2.7 \rfloor &= -3 \\ \lfloor 3.2 \rfloor &= \lfloor 3 \rfloor \\ &= 3\end{aligned}$$

This allows any real number c to be written as;

$$c = \lfloor c \rfloor + (c - \lfloor c \rfloor)$$

This has two properties, $\lfloor c \rfloor$ is obviously always an integer, and $c - \lfloor c \rfloor$ is always **strictly** less than 1.

Assume we have computed a non-integer x^*, and it lives on the boundary of the polytope. We show how to construct a Gomory Cut for the following, where $a_j, b \in \mathbb{R}$ (not always integer);

$$a_1x_1 + \cdots + a_nx_n = b$$

The constraint can then be written as;

$$(\lfloor a_1 \rfloor + \underbrace{(a_1 - \lfloor a_1 \rfloor)}_{f_1})x_1 + \cdots + (\lfloor a_n \rfloor + \underbrace{(a_n - \lfloor a_n \rfloor)}_{f_n})x_n = \lfloor b \rfloor + \underbrace{(b - \lfloor b \rfloor)}_{f}$$

By rearranging terms, we can obtain the following;

$$f_1x_1 + \cdots + f_nx_n - f = \lfloor b \rfloor - \lfloor a_1 \rfloor x_1 - \cdots - \lfloor a_n \rfloor x_n$$

The theorem is as follows, for all $\boldsymbol{x} \in \mathbb{N}_0^n$, satisfying $a_1x_1 + \cdots + a_nx_n = b$;

$$f_1x_1 + \cdots + f_nx_n \geq f$$

Since we know the RHS of the rearranged equation is an integer (all integer coefficients, and all x_i are integers), the LHS must also be an integer. Since $x \geq 0$, $0 \leq f_i \leq 1$;

$$f_1x_1 + \cdots + f_nx_n - f \geq 0 + \cdots + 0 - f > -1$$

Since the LHS can only take integer values, it must be ≥ 0. Therefore, we can say;

$$f_1x_1 + \cdots + f_nx_n - f \geq 0$$

Because we assumed a non-integer solution, we have a row in the optimal simplex tableau containing the following result (the summation is over the non-basic variables), with $y_{i,0} \notin \mathbb{N}_0$;

$$x_i^* + \sum_{j \notin I} y_{i,j}x_k^* = y_{i,0}$$

We can set $f_j = y_{i,j} - \lfloor y_{i,j} \rfloor$ and $f = y_{i,0} - \lfloor y_{i,0} \rfloor$ (and we know that $f > 0$, since it is strictly non-integer). As such, our Gomory Cut is as follows (violated by a non-integer x^*, since $x_j^* = 0$ if $j \notin I$);

$$\sum_{j \notin I} f_jx_j \geq f$$

Worked Example

The following concerns this example;

$$\begin{aligned} \text{maximize} \quad & y = 3x_1 + 4x_2 \\ \text{subject to} \quad & \frac{2}{5}x_1 + x_2 \leq 3 \\ & \frac{2}{5}x_1 - \frac{2}{5}x_2 \leq 1 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{N}_0 \end{aligned}$$

We convert this to a minimisation problem (and scale the coefficients when needed), and add integer slack variables (x_3, x_4);

$$\begin{aligned} \text{minimize} \quad & z = -3x_1 + -4x_2 \\ \text{subject to} \quad & 2x_1 + 5x_2 + x_3 = 15 \\ & 2x_1 - 2x_2 + x_4 = 5 \\ & x_1, x_2, x_3, x_4 \geq 0 \\ & x_1, x_2, x_3, x_4 \in \mathbb{N}_0 \end{aligned}$$

Now solve the LP relaxation (remove the $\in \mathbb{N}_0$ constraint);

BV	x_1	x_2	x_3	x_4	RHS
z	0	0	-1	$-\frac{1}{2}$	$-\frac{35}{2}$
x_2	0	1	$\frac{1}{7}$	$-\frac{1}{7}$	$\frac{10}{7}$
x_1	1	0	$\frac{1}{7}$	$\frac{5}{14}$	$\frac{55}{14}$

Let us generate a cut based on the x_1 row (either works);

$$x_1 + \frac{1}{7}x_3 + \frac{5}{14}x_4 = \frac{55}{14}$$

We compute the f values as follows;

$$\begin{aligned} f_1 &= 1 - \lfloor 1 \rfloor \\ &= 0 \\ f_3 &= \frac{1}{7} \\ f_4 &= \frac{5}{14} \\ f &= \frac{55}{14} - \lfloor \frac{55}{14} \rfloor \\ &= \frac{13}{14} \end{aligned}$$

The Gomory Cut (GC1) is then formulated as;

$$\frac{1}{7}x_3 + \frac{5}{14}x_4 \geq \frac{13}{14} \Rightarrow 2x_3 + 5x_4 \geq 13$$

We add the cut to the LP relaxation, which means to standardise it with an excess variable (note that the previous relaxation solution gives zeroes on the LHS, hence (GC1) becomes infeasible as expected);

$$2x_3 + 5x_4 - x_5 = 13$$

Since we've added a $\geq$ constraint, we now need to perform simplex phase 1, and add an artificial variable ξ_1;

$$2x_3 + 5x_4 - x_5 + \xi_1 = 13$$

This is then repeated until we have an integer solution.

Knapsack Cover Cuts

Consider the following problem, derived from logic about packing problems;

$$3x_1 + 5x_2 + 4x_3 + 2x_4 + 7x_5 \leq 8$$
$$x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}$$

One example is that we both x_2 and x_3 cannot be simultaneously equal to 1, otherwise we'd end up with $5 + 4$ on the RHS, which exceeds the constraint. We add the following cover cut;

$$x_2 + x_3 \leq 1$$

A set S of items in a knapsack problem is called a **cover** (exceeds capacity) if;

$$\sum_{j \in S} w_j > W$$

If S is a cover, then the corresponding knapsack cover cut is;

$$\sum_{j \in S} x_j \leq |S| - 1$$

Usually, we want a minimal cover constraint, a cover constraint such that for all proper subsets T of S;

$$\sum_{j \in T} w_j \leq W$$

Note that for it to be minimal, if we turned off one of the values, it would be feasible again.

Branch and Bound

This divides a problem into smaller parts and prunes out certain branches - we structure our search to scan only through a subset of solutions. Our notation for this is as follows;

- P_i — i^{th} subproblem
- $x^*(P_i)$ — optimal solution to the i^{th} subproblem

The application to MILP is as follows;

1. solve LP relaxation of the problem $P_0 \Rightarrow x^*(P_0)$
2. if this satisfies integrality constraints, we can stop
3. otherwise choose non-integer $x_p^* \in x^*(P_0), p \in Z$
   - **divide** - create two subproblems P_1 and P_2 by adding the constraints $x_p \leq \lfloor x_p^* \rfloor$ and $x_p \geq \lceil x_p^* \rceil$, respectively, to P_0
   - **conquer** - recursively solve the new subproblems, if the optimal solution of the continuous relaxation of P_1 is worse than any known feasible solution for P_0, disregard P_1 (and same for P_2)

Any solution of P_0 which satisfies integrality constraints is feasible in one of P_1 and P_2 - hence we can solve P_0 by solving both P_1 and P_2.