Chapter 1 - Introduction [source: bertsimas 1997]

Q: What is linear programming? Any ideas?

A: The problem of minimizing a linear cost function subject to linear equality and inequality constraints

Lets see what this means:

1.1 Variants of the linear programming problem

Lets start with a concrete example

Example 1.1 The following is a linear programming problem:

Minimize: $2u_1 - u_2 + 4u_3$

Subject to:

$$u_1 + u_2 + u_4 \leq 2 \quad \text{(linear inequality)}$$

$$3u_2 - u_3 = 5 \quad \text{(linear equality)}$$

$$u_3 + u_4 \geq 3 \quad \text{(linear inequality)}$$

$$u_1 \geq 0 \quad \text{(linear inequality)}$$

$$u_3 \leq 0 \quad \text{(linear inequality)}$$

Constraints are of the form:

- $a' u \leq b$
- $a' u = b$
- $a' u \geq b$

Where :

- $a = (a_1, a_2, a_3, a_4)$ is a given vector
- $u = (u_1, u_2, u_3, u_4)$ is the vector of decision variables
- $a'u$ is the inner product: $\sum_{i=1}^{4} a_i u_i$
- b is a given scalar

For example, in the first constraint:

$$u_1 + u_2 + u_4 \leq 2$$

we have $a = (1, 1, 0, 1)$ and $b = 2$

Lets generalize :

In a general Linear programming problem, we are given a $\boxed{\text{cost vector}}$ $c = (c_1, \dots, c_n)$ and we seek to minimize a linear cost function

$$c'u = \sum_{i=1}^{n} c_i u_i$$

Over all n-dimensional vectors $u = (u_1, \dots, u_n)$ subject to a set of linear equality and inequality constraints .

- In particular let M_1, M_2, M_3 be some finite index sets, and suppose that for every i in any of these sets we are given an n-dimensional vector a_i and a scalar b_i, that will be used to form the i^{th} constraint.

- Let also N_1 and N_2 be subsets of $\{1, \dots, n\}$ that indicate which variables u_j are constrained to be nonnegative or nonpositive, respectively.

- We then consider the problem:

$$
\begin{aligned}
\text{minimize:} \quad & c'u \\
\text{subject to:} \quad & a_i'u \geq b_i, \quad i \in M_1 \\
& a_i'u \leq b_i, \quad i \in M_2 \\
& a_i'u = b_i, \quad i \in M_3 \\
& u_j \geq 0, \quad j \in N_1 \\
& u_j \leq 0, \quad j \in N_2
\end{aligned}
$$

Nonnegative: (i.e. all positive decision variables, including zero)

Nonpositive: (i.e. all negative decision variables, including zero)

- Variables $u_1, \dots, u_n$ are called $\boxed{\text{decision variables}}$

- Vector u satisfying all of the constraints is called a $\boxed{\text{feasible solution}}$.

$\boxed{\text{Notes: Solução Viável}}$

- Set of feasible solutions is called the feasible set.

- If j is in neither N_1 nor N_2, there are no restrictions on the sign of x_j, i.e. the variable is $\boxed{\text{free}}$ or $\boxed{\text{unrestricted}}$

- The function $c'x$ is called the $\boxed{\text{objective function}}$ or $\boxed{\text{cost function}}$

- A feasible solution x^* that minimizes the objective function:

$$c' x^* \leq c' x \quad \forall x$$

is called $\boxed{\text{optimal feasible solution}}$ or an $\boxed{\text{optimal solution}}$.

- The value $c'x^*$ is called the optimal cost

- If for every real number K we can find a feasible solution x whose cost is less than K, we say that the optimal cost is $-\infty$ i.e. the cost in unbounded below

Q: But wait, what if we want to study **maximization** instead of minimization?

A: Maximizing $c'u$ is equivalent to minimizing the linear cost function $-c'u$ ☺

Other useful notes:

*1 An equality constraint $a_i'u = b_i$ is equivalent to the two constraints $a_i'u \le b_i$ and $a_i'u \ge b_i$

*2 Any constraint of the form $a_i'u \le b_i$ can be rewritten as $(-a_i)'u \ge -b_i$.

*3 Constraints of the form ~~$a_i'u \ge b_i$~~ $u_j \ge 0$ or $u_j \le 0$ are special cases of constraints of the form $a_i'u \ge b_i$ where a_i is a unit vector ~~and~~ and $b_i = 0$

Question: Why a unit vector?

↳ Answer: Because only one of the variable in u is set to one

Because of *1, *2 and *3:

- We conclude that the $\boxed{\text{feasible set}}$ in a general linear programming problem can be expressed exclusively in terms of inequality constraints of the form $a_i'u \ge b_i$

- Suppose that there is a total of m such constraints indexed by $i = 1, \dots, m$, let $b = (b_1, \dots, b_m)$ and let A be the $m \times n$ matrix whose rows are the row vectors $a_1', \dots, a_m'$ that is

$$A = \begin{bmatrix} a_1' \\ \vdots \\ a_m' \end{bmatrix}$$

Then the $\boxed{\text{constraints}}$ $a_i' u \geq b_i$, $i = 1, \dots, m$ can be expressed compactly in the form $Au \geq b$ and the linear programming problem can be written as

$$\begin{aligned} &\text{minimize: } c'u \\ &\text{subject to: } Au \geq b \end{aligned} \qquad (1.2)$$

$\Downarrow$

$\boxed{\text{Q: What does this all mean?}}$

Example 1.2: The linear programming problem in Example 1.1 can be rewritten as

Minimize: $2u_1 - u_2 + 4u_3$

Subject to:

$$\begin{aligned} -u_1 - u_2 \qquad\qquad - u_4 &\geq -2 \\ 3u_2 \qquad\qquad - u_3 &\geq 5 \\ -3u_2 \qquad\qquad + u_3 &\geq -5 \\ u_3 + u_4 &\geq 3 \end{aligned}$$

$u_1 \geq 0$

$-u_3 \geq 0$

Which is of the same form as the problem (1.2) with $c = (2, -1, 4, 0)$

$$A = \begin{bmatrix} -1 & -1 & 0 & -1 \\ 0 & 3 & -1 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad b' = \begin{pmatrix} -2 \\ 5 \\ -5 \\ 3 \\ 0 \\ 0 \end{pmatrix}$$

Standard form problems

A linear programming problem of the form

Minimize: $c' u$

Subject to: $A u = b$

$u \geq 0$

is said to be in $\boxed{\text{standard form}}$

Q: How can we interpret problems that are in standard form ?

- Suppose that $\underline{u}$ has dimension $\underline{n}$ and let $A_1, \ldots, A_n$ be colums of $\underline{A}$. Then the constraint $\underline{Au = b}$ can be written in the form:

$$\sum_{i=1}^{n} A_i u_i = b$$

Q: Why is this so?

A: Check page

Intuitively:

- There are $\underline{n}$ available resource vectors $A_1, \ldots, A_n$ and a target vector $\underline{b}$
- We wish to "synthesize" the target vector $\underline{b}$ by using a nonnegative amount u_i of each resource vector $\underline{A_i}$
- While minimizing the cost $\sum_{i=1}^{n} c_i u_i$ where $\underline{c_i}$ is the unit cost of the i^{th} resource

$$Au = b$$

$$\begin{array}{cccc} A_1 & A_2 & & A_n \\ \end{array}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

In normal matrix multiplication (i.e. row x column):

$$\begin{cases} a_{11} \cdot u_1 + a_{12} \cdot u_2 + \dots + a_{1n} \cdot u_n = b_1 \\ a_{21} \cdot u_1 + a_{22} \cdot u_2 + \dots + a_{2n} \cdot u_n = b_2 \\ \vdots \\ a_{m1} \cdot u_1 + a_{m2} \cdot u_2 + \dots + a_{mn} \cdot u_n = b_m \end{cases}$$

$$\underbrace{A_1 \cdot u_1 + A_2 \cdot u_2 + \dots + A_n \cdot u_n = b}$$

$$\sum_{i=1}^{n} A_i \cdot u_i = b$$

Example 1.3 (The diet problem)

- Suppose that there are n different foods and m different nutrients, and that we are given the following table with the nutritional content of a unit of each food

	food 1	food 2	...	food n
nutrient 1	a_{11}	a_{12}	...	a_{1n}
nutrient 2	a_{21}	a_{22}	...	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
nutrient m	a_{m1}	a_{m2}	...	a_{mn}

- Let b be a vector with the requirements of an ideal diet
- Problem: Mix nonnegative amounts u_i of the available foods to synthesize the ideal food at minimal cost.

Reduction to Standard Form

- A general linear programming problem can can be transformed into an equivalent problem in standard form

When two problems are equivalent:

- Given a feasible solution to one problem, we can construct a feasible solution to the other, with the same cost.

In particular:

- The two problems have the same optimal cost
- Given an optimal solution to one problem, we can construct an optimal solution to the other.

Q: But how do we transform the problem into another?

A: With a mathematical procedure/algorithm ;)

The problem transformation involves two steps:

(a) Elemination of free variables: Given an unrestricted variable u_j in a problem in general form, we replace it by $u_j^+ - u_j^-$, where u_j^+ and u_j^- are new variables on which we impose the sign constraints $u_j^+ \geq 0$ and $u_j^- \geq 0$

Q: Why?

A: Any real number can be written as the difference of two nonnegative numbers

(b) Elimination of inequality constraints:

Given an inequality constraint of the form:

$$\sum_{j=1}^{n} a_{ij} u_j \leq b_i$$

We introduce a new variable s_i and the standard form constraints:

$$\sum_{j=1}^{n} a_{ij} u_j + s_i = b_i$$

Notes: I.e. we had a certain amount such that the inequality disappear

$$s_i \geq 0$$

Such a variable s_i is called a slack variable

$\therefore_1$ Any general problem can be brought into standard form

$\therefore_2$ We only need methods ~~to develop~~ capable of solving standard form problem

Example 1.4 The problem

Minimize: $2u_1 + 4u_2$

Subject to: $u_1 + u_2 \geq 3$

$3u_1 + 2u_2 = 14$

$u_1 \geq 0$

Is equivalent to the standard form problem:

Notes:

Minimize: $2u_1 + 4u_2^+ - 4u_2^-$ ... Step a)

Subject to: $u_1 + u_2^+ - u_2^- - u_3 = 3$ ... Step a) b)

$3u_1 + 2u_2^+ - 2u_2^- = 14$ ... Step a)

$u_1, u_2^+, u_2^-, u_3 \geq 0$

1.2 Examples of linear programming problems

1.2.1 Production Problem

A firm produces n different goods using m different raw materials.

Problem: Decide how much of each good to produce in order to maximize its total revenue.

1.2.2 Multiperiod planning of electric power capacity

A state wants to plan its electricity capacity for the next T years. The state has a forecast of d_t megawatts. There are two alternatives for electrical generation:

Oil-fired plants: Cost c_t / megawatt, last for 20 yrs

Nuclear plants: Cost n_t / megawatt, last for 15 yrs

Problem: Determine a least cost expansion plan

1.2.3 Scheduling problem

A hospital wants to make a weekly night shift schedule for its nurses. The demand for nurses for the night shift on day j is an integer d_j. Every nurse works 5 days in a row on the night shift.

Problem: Find the minimal number of nurses the hospital needs to hire

Scanned by CamScanner

1.3 Piecewise linear convex objective functions

There is an important class of optimization problems with a $\boxed{\text{nonlinear}}$ objective function that can be cast as linear programming problems:

Definition 1.1

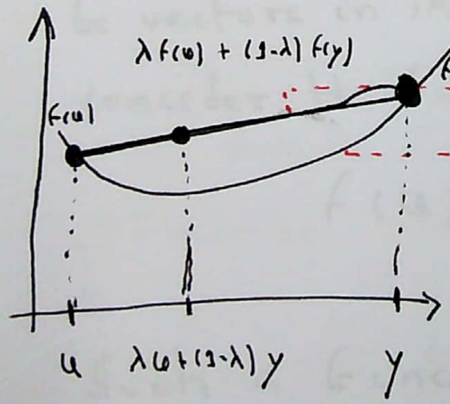
(a) A function $f: \mathbb{R}^n \to \mathbb{R}$ is called $\boxed{\text{convex}}$ if for every $u, y \in \mathbb{R}^n$ and every $\lambda \in [0,1]$ we have:

$$f(\lambda u + (1-\lambda) y) \leq \lambda f(u) + (1-\lambda) f(y)$$

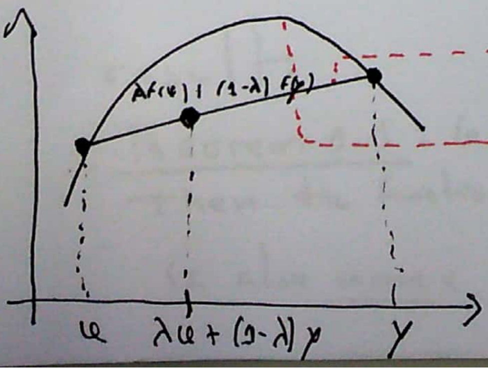
(b) A function $f: \mathbb{R}^n \to \mathbb{R}$ is called $\boxed{\text{concave}}$ if for every $u, y \in \mathbb{R}^n$ and every $\lambda \in [0,1]$ we have

$$f(\lambda u + (1-\lambda) y) \geq \lambda f(u) + (1-\lambda) f(y)$$

Lets see what this means:



Notes:
- Linear function, i.e. a line ‼ ☺
- Convex function



Notes:
- Linear function
- Concave function

- Convex (as well as concave) functions play a central role in optimization.

Also:

- Vector u is a $\boxed{\text{local minimum}}$ of f if $f(u) \le f(y)$ for all y in the vicinity of u.
- Vector u is a $\boxed{\text{global minimum}}$ of f if $f(u) \le f(y)$ for all y.
- A convex function cannot have local minima that fail to be global minima (see previous figure about convex functions). This property is of great help in designing efficient optimization algorithms.
- $\boxed{\text{Piece wise linear convex function}}$: Let $c_1, \dots, c_m$ be vectors in $\mathbb{R}^n$, let $d_1, \dots, d_m$ be scalars and consider the function $f : \mathbb{R}^n \to \mathbb{R}$ defined by

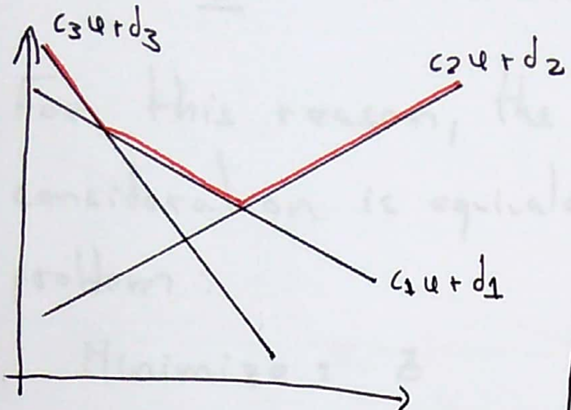
$$f(u) = \max_{i=1,\dots,m} (c_i' u + d_i)$$

Such a function is convex of the following result:

Theorem 1.1: Let $f_1, \dots, f_m : \mathbb{R}^n \to \mathbb{R}$ be convex functions. Then the function f defined by $f(u) = \max_{i=1,\dots,m} f_i(u)$ is also convex.

Q: What does all this mean?

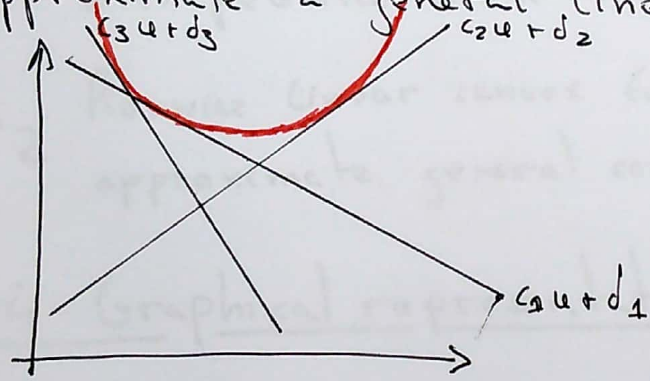
Lets have a look at a visual graphic...



Notes:

- What a mess, right?
- Notice that the <u>red</u> part signals the max
- This max is defined in pieces
- Each piece is a line, i.e. a linear function
- Thus the name piecewise linear convex function

Piecewise linear convex functions can be used to approximate a general linear function:



Notes:

An <u>approximation</u> of a convex function by a piecewise linear convex function

Consider an objective function which is a piecewise linear convex function:

$$\text{Minimize: } \max_{i=1,\dots,m} (c_i' u + d_i)$$

$$\text{Subject to: } Au \geq b$$

- Note that $\max_{i=1,\dots,m} (c_i' u + d_i)$ is equal to the smallest number z that satisfies $z \geq c_i' u + d_i \;\; \forall i$

- For this reason, the optimization problem under consideration is equivalent to the linear programming problem:

$$\begin{aligned} \text{Minimize: } & z \\ \text{Subject to: } & z \geq c_i' u + d_i \quad i = 1, \dots, m \\ & Au \geq b \end{aligned}$$

$\therefore_1$ Linear programming can be used to solve problems with piecewise linear convex cost functions.

$\therefore_2$ Piecewise linear convex function can be used to approximate general convex cost functions

1.4 Graphical representation and solution

Lets see a few simple examples that provide useful geometric insights

Example 1.6

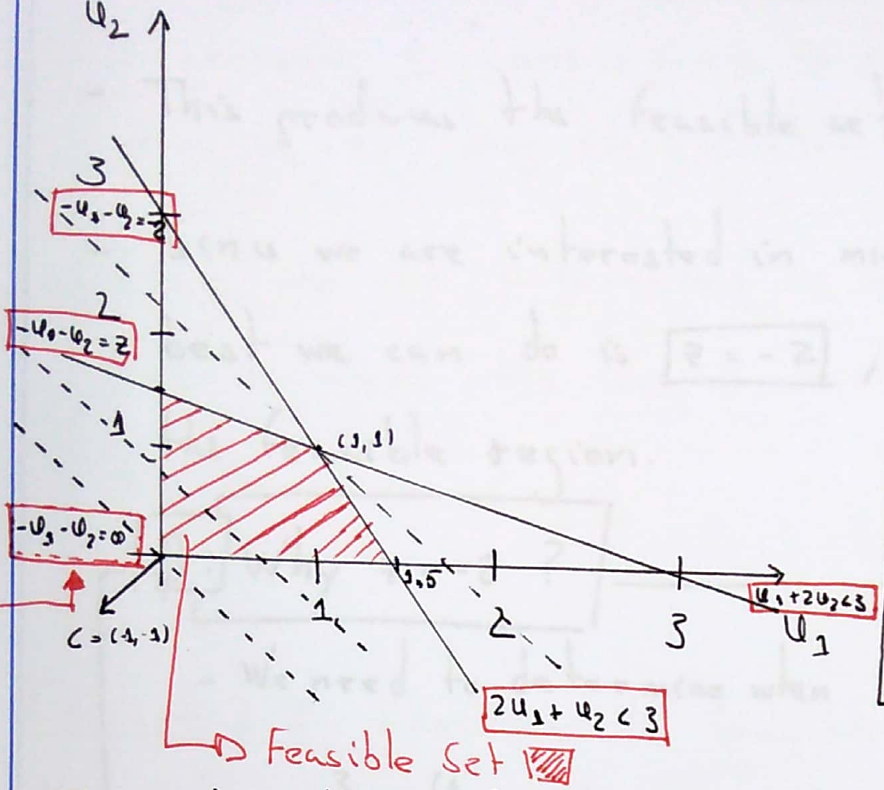
$$\begin{aligned} \text{Minimize: } & -u_1 - u_2 \\ \text{Subject to: } & u_1 + 2u_2 \leq 3 \\ & 2u_1 + u_2 \leq 3 \\ & u_1, u_2 \geq 0 \end{aligned}$$

Minimize: $-u_1-u_2$

subject to: $u_1+2u_2\le 3$

$2u_1+u_2\le 3$

$u_1,u_2\ge 0$



Notes:

- **Equalities:**
  - $-u_1-u_2=0 \Leftrightarrow u_2=-u_1$
  - $u_1+2u_2=3 \Leftrightarrow u_2=\frac{3}{2}-\frac{u_1}{2}$
  - $2u_1+u_2=3 \Leftrightarrow u_2=3-2u_1$

<u>In order to find an optimal solution:</u>

- Consider the set of all points whose cost $c'u$ is equal to z
- This is the line described by equation

$$-u_1-u_2=z$$

- This line should have at least value 0:

$$-u_1-u_2=0$$

- Different values of z lead to different lines, all of them parallel to each other.

Question: Why is this so?

$\left.\begin{array}{l}-u_1-u_2=0\\ -u_1-u_2=1\\ -u_1-u_2=2\end{array}\right\}$ The only thing that is changing is the intersection point with the u_2-axis

- This produces the feasible set (shaded region)

- Since we are interested in minimizing z the best we can do is $\boxed{z = -2}$, without leaving the feasible region.

$\boxed{Q:}$ Why $z = -2$?

- We need to determine when the lines intersect, i.e.:

$$\frac{3}{2} - \frac{u_1}{2} = 3 - 2u_1$$

$$\Leftrightarrow \quad \underset{(2)}{2u_1} - \underset{(1)}{\frac{u_1}{2}} = \underset{(2)}{3} - \underset{(1)}{\frac{3}{2}}$$

$$\Leftrightarrow \quad \frac{4u_1 - u_1}{2} = \frac{6 - 3}{2}$$

$$\Leftrightarrow \quad \frac{\cancel{3}u_1}{\cancel{2}} = \frac{\cancel{3}}{\cancel{2}}$$

$$\Leftrightarrow \quad u_1 = 1$$

- Plugging $u_1 = 1$ into either line equation:

$$\begin{cases} u_2 = \frac{3}{2} - \frac{u_1}{2} \overset{u_1 = 1}{\Leftrightarrow} u_2 = \frac{3}{2} - \frac{1}{2} = 1 \\ u_2 = 3 - 2u_1 \overset{u_1 = 1}{\Leftrightarrow} u_2 = 3 - 2 = 1 \end{cases}$$

- Therefore the intersection point $(u_1, u_2) = (1, 1)$

- If we plug $(u_1, u_2) = (1, 1)$ into the cost

$$-u_1 - u_2 = z$$

$$\Leftrightarrow \quad -1 - 1 = -2$$

- Increasing z corresponds to moving the line $z = -u_1 - u_2$ along the direction of the vector $\underline{c}$

- Since we are interested in minimizing we would like to move the line as much as possible in the direction of $\underline{-c}$

Example 1.8 Consider the feasible set in $\mathbb{R}^2$ defined by constraints:

$$-u_1 + u_2 \leq 1$$

$$u_1 \geq 0$$

$$u_2 \geq 0$$

Notes:

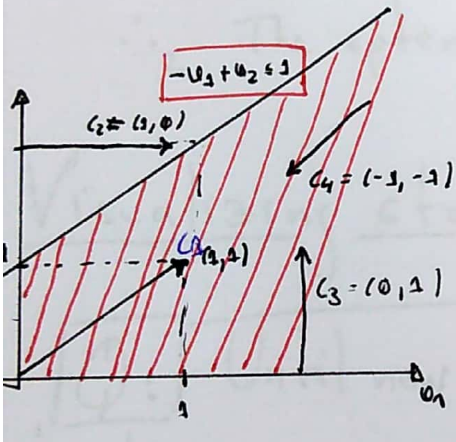
$-u_1 + u_2 = 1$

$\iff -u_2 = 1 - u_1$

$\iff u_2 = u_1 - 1$

Notes:

$-u_1 + u_2 = 1 \iff u_2 = 1 + u_1$



(a) For $c_1 = (1,1)$, $u = (0,0)$ is the unique optimal solution

(b) For $c_2 = (1,0)$, there are multiple optimal solutions namely:

Every vector of the form $u = (0, u_2)$ with $u_2 \in [0, 1]$ is optimal

(c) For $c_3 = (0,1)$, there are multiple optimal solutions namely:

Every vector of the form $u = (u_1, 0)$ with $u_1 \geq 0$

(d) For $c_4 = (-1, -1)$:

- For any feasible solution (u_1, u_2) we can always produce another feasible solution with less cost, by increasing the value of u_1

$\therefore_1$ Therefore no feasible solution is optimal

- By considering vectors (u_1, u_2) with ever increasing values of u_1 and u_2 we can obtain a sequence of feasible solutions whose cost converges to $-\infty$

$\therefore_2$ The optimal cost is $-\infty$.

Visualizing standard form problems

Q: Until now this has been easy, but what happens when the number of dimensions is greater than 3?

Suppose that:

- Matrix A has dimension $m \times n$ (from $A.u \geq b$)

- Decision vector u has dimension n

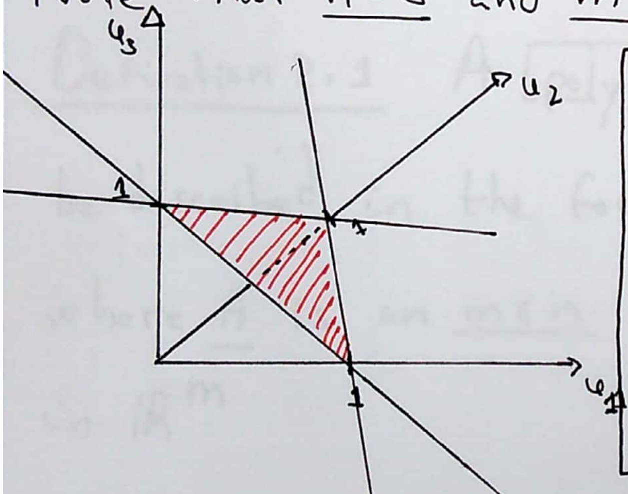
- We have m equality constraints

Consider the feasible set in $\mathbb{R}^3$ defined by the constraints:

$$u_1 + u_2 + u_3 = 1$$

$$u_1, u_2, u_3 \geq 0$$

Note that $n = 3$ and $m = 1$



Notes:

$u_1 u_2$ - trace:

- $u_3 = 0 \Rightarrow u_1 + u_2 = 1 \Leftrightarrow u_1 = 1 - u_2$

$u_1 u_3$ - trace:

- $u_2 = 0 \Rightarrow u_1 + u_3 = 1 \Leftrightarrow u_1 = 1 - u_3$

$u_2 u_3$ - trace:

- $u_1 = 0 \Rightarrow u_2 + u_3 = 1 \Leftrightarrow u_2 = 1 - u_3$

- This triangle can be viewed in a 2D-space if we "stand" on the plane
- Each edge of the triangle corresponds one of the constraints $u_1, u_2, u_3 \geq 0$