MATH 464 – Lecture 1 (01/10/2023)

This is Linear Optimization (CAPSTONE).
I'm Bala Krishnamoorthy, call me Bala.

Today:
* syllabus, logistics
* Optimization in calculus
* Toy example

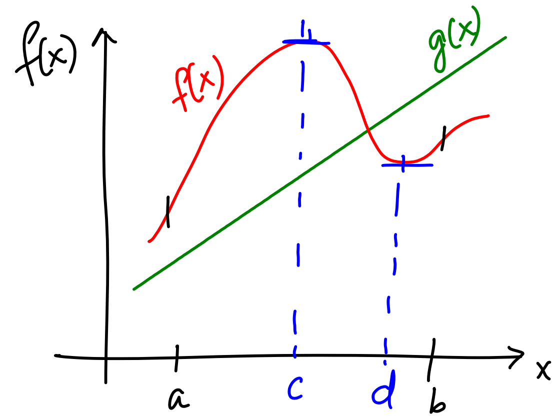
Linear Optimization (or linear programming, LP) is the most basic form of optimization. The word programming is used more commonly than optimization (along with "linear," "non linear," or "integer" etc.) Programming just means a set of instructions (and not computer programming).

LP is used in many areas. My own research includes theory and applications of integer linear programming, or IP in short, where variables are restricted to integers. Related areas I work on include combinatorial optimization, topology, computational biology, geometric measure theory, etc.

This course will place a lot of emphasis on the theory behind LP – so, there will be **many** proof-type exercises in the homework. We will also have a project that will involve the implementation and testing of ~~(some of)~~ the algorithms learned in class.

Optimization in Calculus

Find min/max of $f(x)$ for $a \le x \le b$



min: $f'(x)=0,\ f''(x)>0 \quad x=d$

max: $f'(x)=0,\ f''(x)<0 \quad x=c$

Need to also consider $f(x)$ at the endpoints $x=a$ and $x=b$.

Here, the minimum of $f(x)$ over $[a,b]$ is at $x=a$, and the maximum is at $x=c$.

Instead of $f(x)$, if we consider $g(x)$ which is linear, you need to check only the end points!

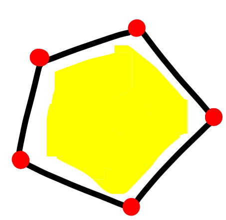
In this course, we will study generalizations of this easier case, i.e., linear function, to higher dimensions

$$\max/\min\ f(x_1,x_2,\dots,x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n = [c_1 \cdots c_n]\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \bar{c}^T\bar{x}$$

subject to linear constraints in $x_1,\dots,x_n$.

My notation:
$\bar{c},\bar{x},\bar{y}$: vectors
x,α,β: 1D variables
A,B: sets/matrices

Things are harder because of the high dimension, but still "easy" because of linearity.



Instead of the closed interval $[a,b]$, we get "polytopes" (generalizations of polygons in 2D).

It turns out we can get away with looking at the corner points, or vertices (just like the end points a,b of $[a,b]$) here!

An aside on optimization classes at WSU

364 - Intro to LP (no proofs)
464 - Linear Optimization (with proofs)
466/566 - Network optimization
564 - Nonlinear Optimization (convex)
565 - Nonconvex Optimization, Optimization in ML.
567 - Integer optimization

Recall from linear algebra (Math 220):

$A\bar{x} = \bar{b}$

Notation: x, y, α, β : single variables
$\bar{x}, \bar{y}$, : vectors (lower case letters with "bar")
A, B, L, U : matrices, sets

$[A \mid \bar{b}] \xrightarrow{\text{EROs}}$ RREF (reduced row echelon form)

elementary row operations

If RREF has a row $[0\ 0 \dots 0 \mid *]$ where $*$ is nonzero, then the system is inconsistent.

Else, the system has
- unique solution
- infinitely many solutions.

We typically work in this setting

From among the infinitely many solutions, find a solution $\bar{x} = \bar{x}^*$ to $A\bar{x} = \bar{b}$ that optimizes $f(\bar{x}) = \bar{c}^T\bar{x}$.

A motivating example

We illustrate the whole pipeline on an example – start with a "word problem", formulate it as a linear program, and also solve it (on the computer).

"Math"!

Dude M. Major's Thursday problem:

Dude has 5 hrs to spend, has two options – party, or get tutored on math. He has \$48 to spend. Here are the costs involved:

Costs:
- tutoring – \$8/hr
- party – \$16/hr

Utility
- tutoring → 2/hr
- party → 3/hr

Dude estimates the utility of each activity in some units. It could be, e.g, in 100s of dollars.

Decide how long Dude will party, and how long he'll get tutored, so that his total utility is maximized.

Decisions to make: How many hours to party, how many hours to get tutored, so that the total utility is maximized?

Let x_1 = # hrs of tutoring and x_2 = # hrs of partying.

$$\begin{array}{lrl} \max & z = 2x_1 + 3x_2 & \text{(total utility)} \\ \text{s.t.} & x_1 + x_2 \leq 5 & \text{(max time)} \\ & 8x_1 + 16x_2 \leq 48 & \text{(max money)} \\ & x_1 \geq 0,\ x_2 \geq 0 & \text{(nonnegativity)} \end{array}$$

Linear Program (LP)

maximize

subject to

The brief "descriptions", e.g., (total utility), are important to help us comprehend what the model is capturing in each constraint and the objective function. Similarly, the sign restrictions, i.e., non-negativity here, are also important.

We solve the problem using the software AMPL, which we will introduce in detail later on.

The "model" (Dude.txt)

```
var x {1..2} >= 0;

maximize TotalUtility: 2*x[1] + 3*x[2];

subject to MaxTime: sum {j in 1..2} x[j] <= 5;
subject to MaxCost: 8*x[1] + 16*x[2] <= 48;
```

Session from AMPL:

```
sw: ampl
ampl: option solver cplex;
ampl: model dude.txt;
ampl: expand MaxTime;
subject to MaxTime:
        x[1] + x[2] <= 5;

ampl: solve;
CPLEX 20.1.0.0: optimal solution;
objective 11
2 dual simplex iterations (1 in phase I)
ampl: display x;
x [*] :=
1  4
2  1
;
```

Dude will get tutoring for 4 hours and party for 1 hour, to get the maximum total utility of 11 units.

We will also learn how AMPL solved the problem, i.e., the mathematics behind the scenes.

MATH 464 – Lecture 2 (01/12/2023)

Today: * general form of LP
* standard form of LP

Homework 1 is posted:
* present arguments that work in general — illustration on small examples is not sufficient

Definitions of Linear Programs (LPs) Forms

General Form of LP:

"is element of"

$$
\begin{aligned}
\min \quad & z = \bar{c}^T \bar{x} \\
\text{s.t.} \quad & \bar{a}_i^T \bar{x} \geq b_i, \quad i \in M_1 \\
& \bar{a}_i^T \bar{x} \leq b_i, \quad i \in M_2 \\
& \bar{a}_i^T \bar{x} = b_i, \quad i \in M_3
\end{aligned}
$$

subsets of indices from $\{1,2,\dots,m\}$

$$
\text{Sign restrictions} \begin{cases} x_j \geq 0, & j \in N_1 \\ x_j \leq 0, & j \in N_2 \\ x_j \text{ urs}, & j \in N_3 \end{cases}
$$

subsets of indices from $\{1,2,\dots,n\}$

Vectors are columns by default (similar to how they are set up in Matlab).

$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ is the vector of decision variables (d.v.'s). Each d.v. x_j is either ≥ 0, ≤ 0 or unrestricted in sign (urs).

Notice $M_1 \cup M_2 \cup M_3 = \{1,2,\dots,m\}$, but $N_1 \cup N_2$ need not be $\{1,2,\dots,n\}$. We could add x_j urs, $j \in N_3$ as a last sign restriction, and then get $N_1 \cup N_2 \cup N_3 = \{1,2,\dots,n\}$.

$\bar{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$ is the objective function coefficients vector.

$\bar{c}^T \bar{x}$ is the objective function.

$\bar{a}_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix}$ is the constraint coefficient vector for the i^{th} constraint.

Stacking the $\bar{a}_i$ vectors as rows gives the $m \times n$ matrix A.

b_i is the right-hand side (rhs) coefficient of i^{th} constraint.

Illustration on Dude's LP:

$$\left.\begin{aligned} \max \quad & z = 2x_1 + 3x_2 && \text{(total utility)} \\ \text{s.t.} \quad & x_1 + x_2 \leq 5 && \text{(max time)} \\ & 8x_1 + 16x_2 \leq 48 && \text{(max money)} \\ & x_1 \geq 0,\ x_2 \geq 0 && \text{(nonnegativity)} \end{aligned}\right\} \text{Linear Program (LP)}$$

max: maximize

s.t.: subject to

"equivalent to"

$$\max \bar{c}^T \bar{x} \equiv \min -\bar{c}^T \bar{x}$$

We could minimize $-\bar{c}^T\bar{x}$ when we have to maximize $\bar{c}^T\bar{x}$.
We could equivalently define the standard form for a maximization LP.

Hence $\bar{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\bar{c} = \begin{bmatrix} -2 \\ -3 \end{bmatrix}$, $\bar{a}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\bar{a}_2 = \begin{bmatrix} 8 \\ 16 \end{bmatrix}$, $\bar{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 48 \end{bmatrix}$

$M_1 = \phi$ (empty set), $M_2 = \{1, 2\}$, $M_3 = \phi$, $N_1 = \{1, 2\}$, $N_2 = \phi$, $N_3 = \phi$.

Def If $\bar{x}$ satisfies all constraints (including sign restrictions), it is called a **feasible solution** or feasible vector.

If $\bar{x}^*$ is a feasible solution such that $\bar{c}^T\bar{x}^* \leq \bar{c}^T\bar{x}$ $\forall$ feasible $\bar{x}$, then $\bar{x}^*$ is an **optimal solution**.

"for all"

With some abuse of notation, we write the general form LP as

$$\begin{aligned} \min \quad & \bar{c}^T\bar{x} \\ \text{s.t.} \quad & A\bar{x} \begin{pmatrix} \geq \\ \leq \\ = \end{pmatrix} \bar{b} \\ & \text{sign restrictions } \bar{x} \end{aligned}$$

We are moving toward the use of results from Linear Algebra on solving $A\bar{x}=\bar{b}$.

With this notation, for the Dude LP, we get

$$A = \begin{bmatrix} 1 & 1 \\ 8 & 16 \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} 5 \\ 48 \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \text{ and } \bar{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Here is another example:

$$\begin{aligned} \min \quad & 2x_1 + 3x_2 - x_3 \\ \text{s.t.} \quad & x_1 + x_2 \geq 4 \\ & 3x_1 - x_2 + 5x_3 \leq 1 \\ & 4x_2 + 3x_3 = 6 \\ & x_1 \geq 0, \; x_2 \leq 0, \; x_3 \text{ urs} \end{aligned}$$

if not specified, a variable is assumed to be urs.

Standard Form of LP

$$\boxed{\begin{array}{ll} \min & \bar{c}^T \bar{x} \\ \text{s.t.} & A\bar{x} = \bar{b} \\ & \bar{x} \geq \bar{0} \end{array}}$$

all variables are non-negative, all constraints are equations.

Recall that we have learned how to solve the general system $A\bar{x} = \bar{b}$. We will use that knowledge to solve any LP, but describe the method for this standard form. Of course, we can convert any general LP to its equivalent standard form LP.

Conversion to Standard Form

1. If $x_j \leq 0$, then replace x_j with $-x_j'$, and $x_j' \geq 0$.

   $\hookrightarrow$ every occurrence of x_j with $-x_j'$

   $-3 \longrightarrow -(3)$ $\rightarrow \geq 0$

   e.g., $x_2 \leq 0 \Rightarrow -x_2 \geq 0$. (we will replace every occurrence of x_2 by $(-x_2')$).

   so, $x_2 \longrightarrow -x_2'$ and add $x_2' \geq 0$

   "replace"

2. If x_j is urs, replace x_j by $x_j^+ - x_j^-$, add $x_j^+, x_j^- \geq 0$.

   $-3 = 0 - 3 \rightarrow x_j^- = 3$ here

   but $-3 = 5 - 8$, for instance $\longrightarrow$ but in an optimal solution, we will have only one of $x_j^+, x_j^- > 0$.

   e.g., x_3 urs $\longrightarrow$ $x_3 \rightarrow x_3^+ - x_3^-$, $x_3^+, x_3^- \geq 0$

x_3^+, x_3^- capture the positive and negative part of x_3. Depending on the value of x_3, only one of x_3^+ and x_3^- will be positive. For instance, if $x_3 = 2$, then $x_3^+ = 2$, $x_3^- = 0$; and if $x_3 = -5$, then $x_3^+ = 0$, $x_3^- = 5$.

The result that both x_i^+ and x_i^- cannot be >0 follows from elementary linear algebra properties. Recall the notion of basic variables (and free or non-basic variables) in the solution of $A\bar{x}=\bar{b}$. The variables that are >0 correspond to basic variables, which in turn correspond to pivot columns of A. But columns of x_i^+ and x_i^- are just (-1) multiples of each other — and hence are linearly dependent. So, both cannot be pivot columns at the same time.

3. If **constraint i is $\geqslant$**, subtract an **excess variable** e_i from the left-hand side (lhs), and add $e_i \geqslant 0$.

e.g., $x_1 + x_2 \geqslant 4 \longrightarrow x_1 + x_2 - e_1 = 4, \; e_1 \geqslant 0$

e_1 captures the amount by which x_1+x_2 exceeds 4. Hence we must insist $e_1 \geqslant 0$. If $e_1 = -2$, for instance, $x_1+x_2=2$, which violates the original constraint.

4. If **constraint i is $\leqslant$**, add **slack variable** s_i to the left-hand side (lhs), and add $s_i \geqslant 0$.

e.g., $3x_1 - x_2 + 5x_3 \leqslant 1$ is replaced by $3x_1 - x_2 + 5x_3 + s_2 = 1,$
$s_2 \geqslant 0.$

We apply these transformations to convert the second LP example to standard form.

↳after Dude's LP.

$$\begin{aligned}
\min \quad & 2x_1 + 3x_2 - \overset{(x_3^+ - x_3^-)}{x_3} \\
\text{s.t.} \quad & x_1 + x_2 - e_1 \geq 4 \\
& 3x_1 - x_2 + 5x_3 + s_2 \leq 1 \\
& 4x_2 + 3x_3 = 6 \\
& x_1 \geq 0, \; \underset{-x_2',\; x_2' \geq 0}{x_2 \leq 0}, \; x_3 \text{ urs}
\end{aligned}$$

$$\begin{aligned}
\min \quad & 2x_1 - 3x_2' - (x_3^+ - x_3^-) \\
\text{s.t.} \quad & x_1 - x_2' - e_1 = 4 \\
& 3x_1 + x_2' + 5(x_3^+ - x_3^-) + s_2 = 1 \\
& -4x_2' + 3(x_3^+ - x_3^-) = 6 \\
& \text{all vars} \geq 0
\end{aligned}$$

(LP in standard form)

Could use x_4, x_5, x_6, etc. instead of x_2', x_3^+, x_3^-, for instance.

Note that we do not have to do anything extra for variables that are ≥ 0 already, and for constraints that are equations.

MATH 464 – Lecture 3 (01/17/2023)

Today: * LP formulations

Download AMPL — follow instructions in Email.

LP Formulations.

→ Chapter 2 in AMPL book (ampl.com)

1. Blending problems – e.g., diet problem

The cost per package of different dinner packages, and the percentage daily value of nutrients (vitamins A, C, B1, B2) per package are given below. The problem is to find the combination of dinner packages to buy so that the requirement of each nutrient for a week are satisfied, i.e., get 700% of each nutrient, and minimize the total cost.

cost per package of dinners:

BEEF	beef	\$3.19
CHK	chicken	2.59
FISH	fish	2.29
HAM	ham	2.89
MCH	macaroni & cheese	1.89
MTL	meat loaf	1.99
SPG	spaghetti	1.99
TUR	turkey	2.49

Percentage daily values of nutrients per dinner package

	A	C	B1	B2
BEEF	60%	20%	10%	15%
CHK	8	0	20	20
FISH	8	10	15	10
HAM	40	40	35	10
MCH	15	35	15	15
MTL	70	30	15	15
SPG	25	50	25	15
TUR	60	20	15	10

Decision variables (d.v.'s)

Let x_j = # dinner packages of type j, j = BEEF, CHK, …, TUR

Or, you could say, x_j = # dinner … , for $j = 1, \dots, 8$, $1 \equiv$ BEEF, …, $8 \equiv$ TUR.

It is important to define the d.v.'s clearly. In particular, statements of the form "Let x_{BEEF} = BEEF" would not cut it!

Objective Function

it's fine to use dots in this manner

$$\underset{\text{minimize}}{\min} \quad 3.19\,x_{BEEF} + 2.59\,x_{CHK} + \cdots + 2.49\,x_{TUR} \qquad (\text{total cost})$$

Constraints

Weekly requirement = 7x100

"subject to"

$$\begin{aligned}
\text{s.t.} \quad & 60\,x_{BEEF} + 8\,x_{CHK} + \cdots + 60\,x_{TUR} \geq 700 && (\text{weekly Vit A}) \\
& 20\,x_{BEEF} + \cdots + 20\,x_{TUR} \geq 700 && (\text{Weekly Vit C}) \\
& 10\,x_{BEEF} + \cdots + 15\,x_{TUR} \geq 700 && (\text{weekly Vit B1}) \\
& 15\,x_{BEEF} + \cdots + 10\,x_{TUR} \geq 700 && (\text{'' '' B2})
\end{aligned}$$

Sign restrictions

$$x_j \geq 0 \quad \forall j \qquad (\text{non-neg})$$

Here is the entire LP:

$$\begin{aligned}
\min \quad & 3.19\,x_{BEEF} + 2.59\,x_{CHK} + \cdots + 2.49\,x_{TUR} && (\text{total cost}) \\
\text{s.t.} \quad & 60\,x_{BEEF} + 8\,x_{CHK} + \cdots + 60\,x_{TUR} \geq 700 && (\text{weekly Vit A}) \\
& 20\,x_{BEEF} + \cdots + 20\,x_{TUR} \geq 700 && (\text{'' '' C}) \\
& 10\,x_{BEEF} + \cdots + 15\,x_{TUR} \geq 700 && (\text{B1}) \\
& 15\,x_{BEEF} + \cdots + 10\,x_{TUR} \geq 700 && (\text{'' '' B2}) \\
& x_j \geq 0 \quad \forall j && (\text{non-neg})
\end{aligned}$$

As we get more familiar and comfortable with LP formulations, we can jump directly to writing the final LP, and skip writing many of the intermediate steps in detail.

We solve the problem in AMPL. The model and data files are available from ampl.com (they're more general than what we need, as they consider some variations).

Model file:

```
set NUTR;
set FOOD;

param cost {FOOD} > 0;
param f_min {FOOD} >= 0;
param f_max {j in FOOD} >= f_min[j];

param n_min {NUTR} >= 0;
param n_max {i in NUTR} >= n_min[i];

param amt {NUTR,FOOD} >= 0;

var Buy {j in FOOD} >= f_min[j], <= f_max[j];

minimize Total_Cost:  sum {j in FOOD} cost[j] * Buy[j];

subject to Diet {i in NUTR}:
   n_min[i] <= sum {j in FOOD} amt[i,j] * Buy[j] <= n_max[i];
```

Data file (the actual numbers):

```
set NUTR := A B1 B2 C ;
set FOOD := BEEF CHK FISH HAM MCH MTL SPG TUR ;

param:   cost  f_min  f_max :=
  BEEF   3.19    0     100
  CHK    2.59    0     100
  FISH   2.29    0     100
  HAM    2.89    0     100
  MCH    1.89    0     100
  MTL    1.99    0     100
  SPG    1.99    0     100
  TUR    2.49    0     100 ;

param:   n_min  n_max :=
   A      700   10000
   C      700   10000
   B1     700   10000
   B2     700   10000 ;

param amt (tr):
          A    C   B1   B2 :=
   BEEF   60   20   10   15
   CHK     8    0   20   20
   FISH    8   10   15   10
   HAM    40   40   35   10
   MCH    15   35   15   15
   MTL    70   30   15   15
   SPG    25   50   25   15
   TUR    60   20   15   10 ;
```

AMPL session:

```
ampl: option solver cplex;
ampl: model "c:/Program
Files/AMPL/ampl_mswin64/models/diet.mod";
ampl: data "c:/Program Files/AMPL/ampl_mswin64/models/diet.dat";
ampl: solve;
CPLEX 20.1.0.0: optimal solution; objective 88.2
1 dual simplex iterations (0 in phase I)

ampl: display Buy;
Buy [*] :=
BEEF   0
 CHK   0
FISH   0
 HAM   0
 MCH  46.6667
 MTL   0
 SPG   0
 TUR   0
;

ampl:
```

It appears we want to buy just mac 'n' cheese, as it's cheapest, and has enough of each of the four vitamins being studied!

We could consider several variations easily to this basic model.

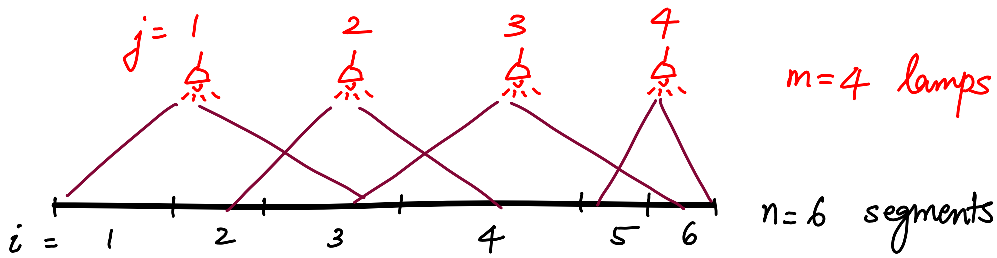
(3.4)

→ Bertsimas-Tsitsiklis: Intro to Linear Opt.

2. Lighting Problem (BT-ILO Prob 1.8):

Exercise 1.8 (Road lighting) Consider a road divided into n segments that is illuminated by m lamps. Let p_j (d.v.s) be the power of the jth lamp. The illumination I_i (d.v.s) of the ith segment is assumed to be $\sum_{j=1}^{m} a_{ij} p_j$, where a_{ij} (data) are known coefficients. Let I_i^* (data) be the desired illumination of road i.

We are interested in choosing the lamp powers p_j so that the illuminations I_i are close to the desired illuminations I_i^* (data). Provide a reasonable linear programming formulation of this problem. Note that the wording of the problem is loose and there is more than one possible formulation.



The a_{ij} values (given data) captures all the information about whether lamp j illuminates segment i, and by how much. For instance, lamp 4 in the illustration above does not illuminate segment 2 at all, and hence $a_{24}=0$. Similarly, $a_{41}=0$ as well.

The main d.v.'s are specified in the problem itself — p_j and I_i. We will add some more d.v.'s in order to write the formulation(s).

Let p_j = power of lamp j, $j=1,\ldots,m$

I_i = total illumination of road segment i, $i=1,\ldots,n$.

<u>Interpretation 1</u>

* Illumination I_i must be at least I_i^*;
* Given the above condition is met, we want to minimize excess illumination, or
* minimize total power spent.

Let e_i = excess illumination in Segment i $(e_i = I_i - I_i^*)$

extra d.v.'s

$$\min \sum_{i=1}^{n} e_i \qquad \text{(total excess illumination)}$$

or

$$\min \sum_{j=1}^{m} p_j \qquad \text{(total power)}$$

s.t.

$$I_i \geq I_i^*, \quad i=1,\dots,n \quad \text{(min. required illum.)}$$

$$I_i = \sum_{j=1}^{m} a_{ij} p_j, \quad i=1,\dots,n \quad \text{(illumination of segment } i\text{)}$$

$$e_i = I_i - I_i^*, \quad i=1,\dots,n \quad \text{(excess illumination)}$$

$$\text{all vars} \geq 0 \qquad \text{(non-neg)}$$

$e_i \geq 0 \implies I_i - I_i^* \geq 0$

$e_i \geq 0 \implies I_i \geq I_i^*$

So we could avoid the first set of constraints. But it's better to leave them in for improved readability.

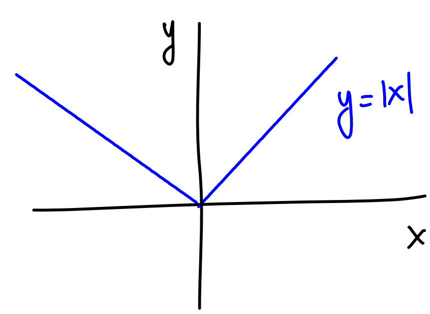
<u>Interpretation 2:</u> Get I_i "as close as possible" to I_i^*.

In other words, we want to minimize sum of $|I_i - I_i^*|$ $\forall i$.

absolute value

$y = |x|$ is nonlinear, but is "quite close to being linear"

$|x|$ is a piecewise linear (PL) function $\longrightarrow$ more on modeling PL functions later.



3. Prob 1.11 (pg 36 BT-ILO): Currency conversion

Exercise 1.11 (Optimal currency conversion) Suppose that there are ~~N~~ n available currencies, and assume that one unit of currency i can be exchanged for r_{ij} units of currency j. (Naturally, we assume that $r_{ij} > 0$.) There also certain regulations that impose a limit u_i on the total amount of currency i that can be exchanged on any given day. Suppose that we start with ~~B~~ b units of currency 1 and that we would like to maximize the number of units of currency ~~N~~ n that we end up with at the end of the day, through a sequence of currency transactions. Provide a linear programming formulation of this problem. Assume that for any sequence $i_1, \ldots, i_k$ of currencies, we have $r_{i_1 i_2} r_{i_2 i_3} \cdots r_{i_{k-1} i_k} r_{i_k i_1} \leq 1$, which means that wealth cannot be multiplied by going through a cycle of currencies.

We prefer to use n, b, etc. instead of N, B: it's just our notation convention!

n currencies

r_{ij} exchange rates

u_i max amount of currency i that can be exchanged

b amount of currency 1 at start

Assumption about r_{ij}'s: Cannot cycle through currencies to make money!

e.g. $\$ \longrightarrow ¥ \longrightarrow € \longrightarrow \$$

yen Euro

$r_{\$¥} = 100$

$r_{¥€} = \cancel{0.05}$ 0.005

$r_{€\$} = \cancel{2.5}$ 1.5

1\$ ⟶ 100¥ ⟶ 5 € ⟶ 12.5\$ ✗

↳ 0.5 € ⟶ 0.75\$ ✓

Because of this assumption, it is clear that we will exchange any currency i to currency j only once in a day. If we convert a currency i to j more than once in a day, we will lose — might as well make all $i \to j$ conversions in one go!

We must do some exchanges for sure, as we want to convert (up to) b units of currency 1 to as many units of currency n as possible.

Let x_{ij} = # units of currency i exchanged to currency j
$i \neq j$, $i \in \{1,2,\dots,n\}$, $j \in \{1,2,\dots,n\}$.

Alternatively, we could restrict the index set of i (first index) to $\{1,2,\dots,n-1\}$ — we want to maximize the amount of currency n in the end, so an optimal solution would set $x_{nj} = 0 \ \forall j$ any way!

We'll finish the formulation in the next class...

MATH464 – Lecture 4 (01/19/2023)

Today:
* convex functions
* piecewise linear (PL) convex functions
* PL convex functions in LP

We first finish the formulation instance from last lecture.

3. Currency Exchange LP (continued...)

$$\max \sum_{i=1}^{n-1} x_{in} \qquad \text{(total \# units of currency } n\text{)}$$

$$\text{s.t.} \quad \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij} \le u_i, \quad i=1,\dots,n-1 \qquad \text{(max exchange of currency } i\text{)}$$

same!

$$\sum_{j=2}^{n} x_{1j} \le b \qquad \text{(amt of currency 1 at start)}$$

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij} \le \underbrace{\sum_{\substack{k=1 \\ k\neq i}}^{n-1} r_{ki} x_{ki}}_{\text{max \# currency } i \text{ we can get by converting any other currency to currency } i.} \quad i=2,\dots,n-1 \qquad \left(\text{max amt of currency } i \text{ that can be exchanged}\right)$$

$$x_{ij} \ge 0 \quad \forall\, i,j \qquad \text{(non-negativity).}$$

The limits of b and u_1 are **Both** applied to the total amount of Currency 1 that can be exchanged. The two Bounds are independently imposed. The third set of constraints is the corresponding Bound for Currencies 2 to $n-1$ (similar to the Bound b on currency 1).

Piecewise linear (PL) Convex Functions

→ a class of otherwise nonlinear functions that could still be modeled using linear functions!

We first define convex functions.

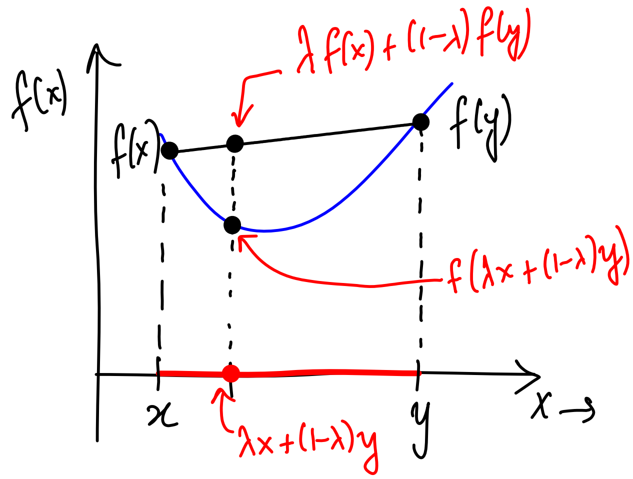
Def A function $f: \mathbb{R}^n \to \mathbb{R}$ is **convex** if $\forall \bar{x}, \bar{y} \in \mathbb{R}^n$ and $\forall \lambda \in [0,1]$

$$f(\lambda \bar{x} + (1-\lambda)\bar{y}) \leq \lambda f(\bar{x}) + (1-\lambda) f(\bar{y}) \qquad (*)$$

$\geq$ → f is concave

$\lambda \bar{x} + (1-\lambda)\bar{y}$ for $\lambda \in [0,1]$ is any point in the line segment joining $\bar{x}$ and $\bar{y}$.

Illustration in 1D:



The graph of $f(\bar{x})$ lies at or below ($\leq$) the line segment connecting $f(\bar{x})$ and $f(\bar{y})$.

If $f(\bar{x})$ is linear, we get $=$ (in place of $\leq$) in $(*)$, i.e., a linear function is both convex and concave.

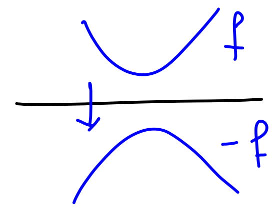
Def A **linear function** $f: \mathbb{R}^n \to \mathbb{R}$ is specified as

$$f(\bar{x}) = \bar{a}^T \bar{x} = \sum_{i=1}^{n} a_i x_i, \quad a_i \in \mathbb{R} \ \forall i \ (\text{or } \bar{a} \in \mathbb{R}^n).$$

Def An **affine function** $f : \mathbb{R}^n \to \mathbb{R}$ is specified as

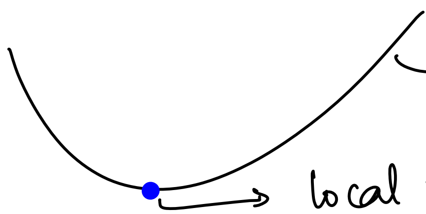
$$f(\bar{x}) = a_0 + \bar{a}^T \bar{x} = a_0 + \sum_{i=1}^{n} a_i x_i \,, \quad a_i \in \mathbb{R} \ \forall i.$$

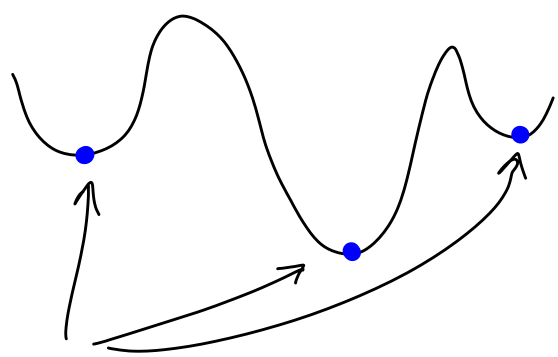
Note f is convex iff $-f$ is concave. Multiplying $(*)$ by -1 flips the $\le$ to $\ge$, giving that $-f(\cdot)$ satisfies the condition for a function being concave. → Intuitively, $-f$ is obtained by the mirror reflection of f across the x-axis.



Why study convex functions?

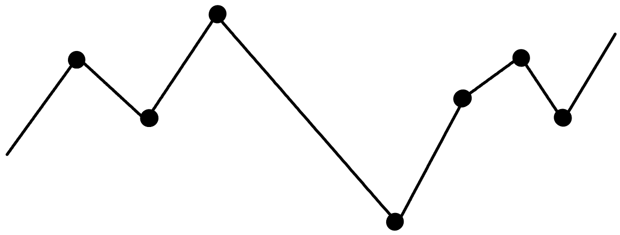
Convex functions have unique global minima!





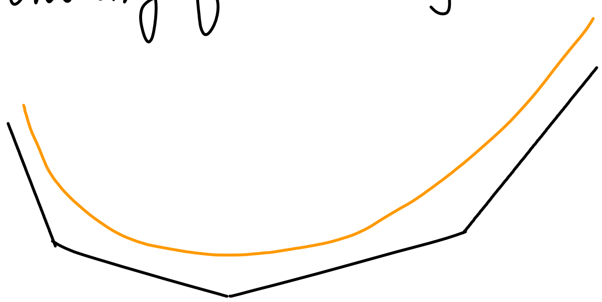
But then again, we are studying *linear* programming, and all the curves drawn so far look non-linear!?

We want to study piecewise linear (PL) functions, which consist of several pieces, each of which is linear.



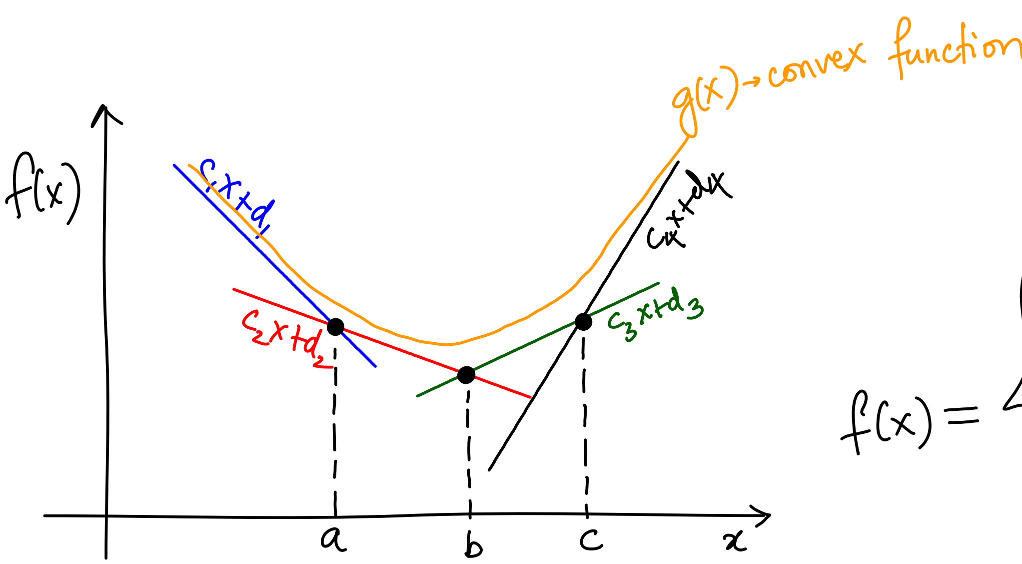
→ not convex or concave!

We will study piecewise linear convex functions, which have the following form. They could also be used to approximate nonlinear convex functions, as shown in orange.



We can explicitly define a PL convex function in 1D by specifying the expression for each linear piece as follows:

Note: the general such function could have m pieces, not necessarily 4 all the time!



$$f(x) = \begin{cases} c_1x+d_1, & x \le a \\ c_2x+d_2, & a \le x \le b \\ c_3x+d_3, & b \le x \le c \\ c_4x+d_4, & c \le x \end{cases}$$

A more compact way of stating $f(x)$ here is as follows:

$$f(x) = \max_{i=1,\dots,4} \{c_i x + d_i\}$$

replacing max with min will not give a concave function here! But,

$$h(x) = \min_{i=1,\dots,4} \{-(c_i x + d_i)\}$$ is a PL concave function.

We could extend this specification to higher dimensions to define PL convex functions in general as follows:

Def $f : \mathbb{R}^n \to \mathbb{R}$ is a piecewise-linear (PL) convex function if

$$f(\bar{x}) = \max_{i=1,\dots,m} (\bar{c}_i^T \bar{x} + d_i) \quad \text{for } \bar{c}_i \in \mathbb{R}^n, d_i \in \mathbb{R} \ \forall i.$$

We can prove that $f(\bar{x})$ is indeed convex. Notice that each piece, i.e., $\bar{c}_i^T \bar{x} + d_i$ is affine, and hence convex. We prove a more general result — the max of a set of convex functions is convex.

Recall, $f(\bar{x})$ is convex if $f(\lambda \bar{x} + (1-\lambda)\bar{y}) \leq \lambda f(\bar{x}) + (1-\lambda) f(\bar{y}) \ \forall \bar{x}, \bar{y} \in \mathbb{R}^n, \lambda \in [0,1]$.

This will be the first proof-type problem for us. We will see several such proofs — but they should be easier than many proofs in analysis, for instance!

Theorem 1.1 (BT-ILO) Let $f_1, \dots, f_m : \mathbb{R}^n \to \mathbb{R}$ be convex functions. Then $f(\bar{x}) = \max\limits_{i=1,\dots,m} f_i(\bar{x})$ is convex.

Proof Let $\bar{x}, \bar{y} \in \mathbb{R}^n$ and $\lambda \in [0,1]$.

We need to show $f(\lambda\bar{x} + (1-\lambda)\bar{y}) \leq \lambda f(\bar{x}) + (1-\lambda) f(\bar{y})$.

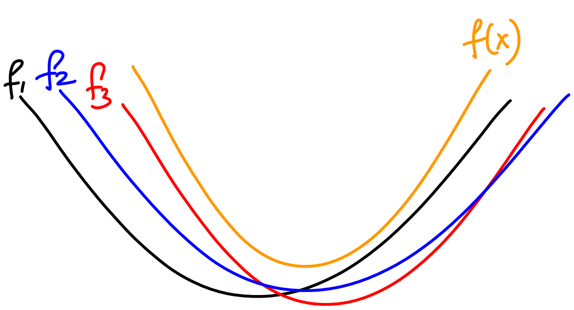
$$
\begin{aligned}
f(\lambda\bar{x}+(1-\lambda)\bar{y}) &= \max_{i=1,\dots,m} f_i(\lambda\bar{x} + (1-\lambda)\bar{y}) \\
&\leq \max_{i=1,\dots,m} \left(\lambda f_i(\bar{x}) + (1-\lambda) f_i(\bar{y})\right), \text{ as each } f_i \text{ is convex} \\
&\leq \max_{i=1,\dots,m} \lambda f_i(\bar{x}) + \max_{i=1,\dots,m} (1-\lambda) f_i(\bar{y}) \quad \to \text{as } \max_i (a_i+b_i) \leq \max_i (a_i) + \max_i (b_i) \\
&= \lambda \max_{i=1,\dots,m} f_i(\bar{x}) + (1-\lambda) \max_{i=1,\dots,m} f_i(\bar{y}) \quad \text{as both } \lambda, 1-\lambda \geq 0 \\
&= \lambda f(\bar{x}) + (1-\lambda) f(\bar{y})
\end{aligned}
$$

this is a crucial requirement!

i.e., $f(\lambda\bar{x} + (1-\lambda)\bar{y}) \leq \lambda f(\bar{x}) + (1-\lambda) f(\bar{y})$.

QED or $\square$

The result works for nonlinear convex functions as well:



QED: quod erat demonstrandum: Latin for "that which has to be demonstrated". Specifies the end of proof. Some journals & books use ■ or □ instead.

PL convex functions in LP

Consider the following generalization of an LP:

$$\min \left\{ \max_{i=1,\dots,m} \left(\bar{c}_i^T \bar{x} + d_i \right) \right\} \longrightarrow \text{PL convex function}$$

$$\text{s.t.} \quad A\bar{x} \geq \bar{b} \longrightarrow \text{sign restrictions are included here.}$$

We can model this problem as an LP:

$$\boxed{\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & z \geq \bar{c}_i^T \bar{x} + d_i, \quad i=1,\dots,m \\ & A\bar{x} \geq \bar{b} \end{aligned}} \quad \text{— (LP)}$$

z is a single urs variable. We specify that z is $\geq$ each piece. Then when we minimize z, we get the desired model!

Note that (the optimal) z could indeed be < 0 here, depending on how the m pieces sit!

What about PL convex functions in constraints?

If we have $f(\bar{x}) \leq h$ where $f(\bar{x}) = \max_{i=1,\dots,m} (\bar{f}_i^T \bar{x} + g_i)$,

then we can replace this constraint by m linear inequalities:

$$\bar{f}_i^T \bar{x} + g_i \leq h, \quad \text{for } i=1,\dots,m.$$

Once again, the logic is similar. To specify that the largest of m functions is $\leq h$, we instead specify each of the m pieces is $\leq h$. Then the largest of them will also be $\leq h$.

MATH464 – Lecture 5 (01/24/2023)

Today:
* Absolute values in LP
* Two LP formulations (w/ absolute value terms)
* Solving LPs in 2D

Recall

$$\min \max_{i=1,\dots,m} (\bar{c}_i^T \bar{x} + d_i) \quad \text{s.t.} \quad A\bar{x} \geq \bar{b}$$

$$\boxed{\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & z \geq \bar{c}_i^T \bar{x} + d_i, \; i=1,\dots,m \\ & A\bar{x} \geq \bar{b} \end{aligned}}$$

The intuition one wants to have (for the objective function) is the following. $\max_{i=1,\dots,m} \{\bar{c}_i^T\bar{x}+d_i\}$ represents the surface of a PL "vessel opened upwards". When we compute $\min \max_{i=1,\dots,m} \{\bar{c}_i^T\bar{x}+d_i\}$, we are pushing a blanket or a piece of cloth down on this vessel. We will go down and hit one of the linear pieces. But if we are working with $\max \max_{i=1,\dots,m} \{\bar{c}_i^T\bar{x}+d_i\}$, we would be pulling the blanket up, while the restraining vessel is below it. As such, we could pull the blanket up as much as we want – in fact, the blanket will no longer model the max of the m pieces – it'll just be pulled up without limit!

Similarly if we want to model $\min_{i=1,\dots,m} f_i(\bar{x}) \leq h$, we do not want to write $f_i(\bar{x}) \leq h \; \forall i$. We only want the smallest of the m functions $f_i(\bar{x})$ to be $\leq h$, while the others could indeed be $> h$. Of course, if each $f_i(\bar{x}) \leq h$, then so will be the smallest of them. But we are putting too much restriction here!

Indeed, we would need extra binary variables to model these situations correctly!

LPs with absolute values

We now consider some direct applications of these ideas. Consider the following optimization problem, which is not an LP as written.

$$\min \sum_{i=1}^{n} c_i |x_i| \quad \text{with } c_i \geqslant 0$$
$$\text{s.t.} \quad A\bar{x} \geqslant \bar{b}$$

Recall that $|x_i| = \max\{x_i, -x_i\}$.

If $c_i < 0$, then we have the situation of $\max\ c_i \max\{x_i, -x_i\}$, which cannot be easily modeled.

We consider two LP formulations for this problem.

1. $$\min \sum_{i=1}^{n} c_i z_i$$ z_i models $|x_i|$
$$\text{s.t.} \quad \left.\begin{array}{l} z_i \geqslant x_i \\ z_i \geqslant -x_i \end{array}\right\} i = 1, \dots, m$$ two constraints imply $z_i \geqslant 0$
$$A\bar{x} \geqslant \bar{b}$$

Notice that this approach mirrors the use of z previously to model $\min \max_{i=1,\dots,m} \{\bar{c}_i^T \bar{x} + d_i\}$.

2. Using the idea of $x^{\pm}$ to model x urs:

$$\min \sum_{i=1}^{n} c_i (x_i^+ + x_i^-)$$
$$\text{s.t.} \quad A(\bar{x}^+ - \bar{x}^-) \geqslant \bar{b}$$
$$\bar{x}^+, \bar{x}^- \geqslant \bar{0}$$

$$\bar{x}^+ = \begin{bmatrix} x_1^+ \\ x_2^+ \\ \vdots \\ x_n^+ \end{bmatrix} \qquad \bar{x}^- = \begin{bmatrix} x_1^- \\ \vdots \\ x_n^- \end{bmatrix}$$

x_i^+, x_i^-: model the positive/negative part of x_i — in the optimal solution, only one of them could be > 0. Both could be 0 (modeling $x_i = 0$).

$c_i \geqslant 0$ forces only one of x_i^+, x^- to be > 0, if at all.

Say $x_i = 3$. We could write $x_i = 0 - 3$ or $x_i = 2 - 5$ or even $x_i = 2023 - 2026$. The pairs (x_i^+, x_i^-) are $(0,3), (2,5)$, or $(2023, 2026)$. Each x_i^+ and x_i^- is $\geqslant 0$. Among all such pairs, we want to pick one with the smallest $x_i^+ + x_i^-$ sum. Indeed, we pick $(0,3)$! With $c_i \geqslant 0$, we get the same result when we want to minimize $c_i(x_i^+ + x_i^-)$.

Example: Write an LP formulation for this problem.

$$\begin{aligned} \min \quad & 2x_1 + |x_2| \\ \text{s.t.} \quad & 5x_1 + 3x_2 \geq 4 \\ & |x_1| + x_2 \leq 3 \\ & x_1, x_2 \text{ urs} \end{aligned}$$

We have $|x_2| = \max\{x_2, -x_2\}$, and

$$|x_1| + x_2 \leq 3$$

"equivalent to" $\equiv \max\{x_1, -x_1\} + x_2 \leq 3$

$$\equiv \max\{x_1 + x_2, -x_1 + x_2\} \leq 3$$

Here is the LP formulation, using the ideas we just introduced:

$$\begin{aligned} \min \quad & 2x_1 + z_2 \\ \text{s.t.} \quad & z_2 \geq x_2 \\ & z_2 \geq -x_2 \\ & 5x_1 + 3x_2 \geq 4 \\ & x_1 + x_2 \leq 3 \\ & -x_1 + x_2 \leq 3 \end{aligned}$$

Notice that z_2 models $|x_2|$ (the coefficient $c_2 = 1$ here).

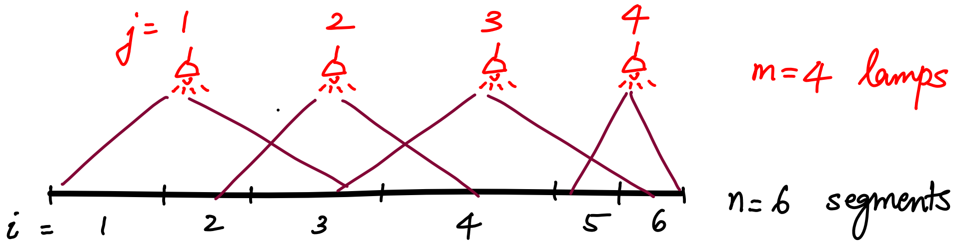
Since x_1, x_2 are urs to start with, we do not mention any sign restrictions here. Even though z_2 is not explicitly declared to be ≥ 0, the first two constraints ensure $z_2 \geq 0$.

2. Back to Lighting Problem (BT-ILO Prob 1.8):

d.V.s

Exercise 1.8 **(Road lighting)** Consider a road divided into n segments that is illuminated by m lamps. Let p_j be the power of the jth lamp. The illumination I_i d.V.s of the ith segment is assumed to be $\sum_{j=1}^{m} a_{ij} p_j$, where a_{ij} are known coefficients. data Let I_i^* data be the desired illumination of road i.

data We are interested in choosing the lamp powers p_j so that the illuminations I_i are close to the desired illuminations I_i^*. Provide a reasonable linear programming formulation of this problem. Note that the wording of the problem is loose and there is more than one possible formulation.



Interpretation 2 Get "as close as possible" to I_i^* (could be above or below).

Minimize total deviations from I_i^*'s.

d.V.'s p_j, I_i, e_i ← The e_i's here model $|I_i - I_i^*|$, and not excess.

$$\min \sum_{i=1}^{n} e_i \qquad \text{(total deviation from } I_i^*\text{)}$$

$$\text{s.t.} \quad \left.\begin{aligned} e_i &\geq I_i - I_i^* \\ e_i &\geq I_i^* - I_i \end{aligned}\right\} \quad \text{(absolute value of } I_i - I_i^*\text{)}$$

$$I_i = \sum_{j=1}^{n} a_{ij} p_j, \quad i = 1, \dots, n \quad \text{(illumination in segment } i\text{)}$$

$$p_j \geq 0 \;\; \forall j \quad \text{(non-neg)}$$

$e_i \geq 0$ is implied by these constraints.

BT-1LD **Exercise 1.10 (Production and inventory planning)** A company must deliver d_i units of its product at the end of the ith month. Material produced during a month can be delivered either at the end of the same month or can be stored as inventory and delivered at the end of a subsequent month; however, there is a storage cost of c_1 dollars per month for each unit of product held in inventory. The year begins with zero inventory. If the company produces x_i units in month i and x_{i+1} units in month $i+1$, it incurs a cost of $c_2|x_{i+1} - x_i|$ dollars, reflecting the cost of switching to a new production level. Formulate a linear programming problem whose objective is to minimize the total cost of the production and inventory schedule over a period of twelve months. Assume that inventory left at the end of the year has no value and does not incur any storage costs. assume ≥ 0!

d_i = demand at end of month i,

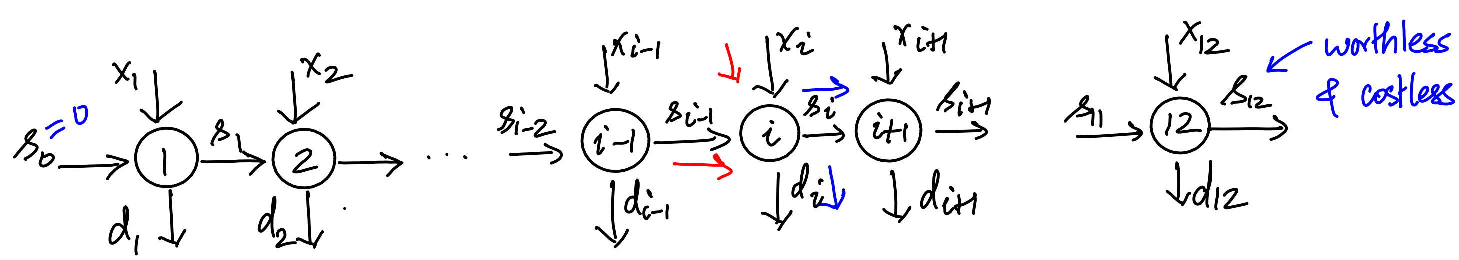
C_1 = storage cost,

x_i = # units produced in month i, $i = 0, 1, \dots, 12$ (d.v.'s),

$C_2|X_{i+1} - X_i|$ = cost of switching production levels.

Let s_i = # units in inventory at end of month i, $i = 0, 1, \dots, 12$ (d.v.'s).

Here is a visualization of the set up:



Node i represents the "flow" of units in month i.
We want "flow balance", i.e., total inflow = total outflow.

$$\underbrace{s_{i-1} + x_i}_{\text{inflow}} = \underbrace{s_i + d_i}_{\text{outflow}}, \quad i = 1, \dots, 12.$$

Goal: minimize $\sum_{i=1}^{11} c_1 s_i + \sum_{i=0}^{11} c_2 \underbrace{|x_{i+1} - x_i|}_{z_i \text{ to model}}$

We need another set of d.v.s, say z_i, to model $|x_{i+1} - x_i|$. We use the same technique as used before to model absolute value terms in min-objective functions.

Here's the full LP:

$$\min \quad \sum_{i=1}^{11} c_1 s_i + \sum_{i=0}^{11} c_2 z_i$$

s.t.

$$\left.\begin{array}{l} z_i \geq x_{i+1} - x_i \\ z_i \geq x_i - x_{i+1} \end{array}\right\} \text{ for } i = 0, \ldots, 11 \quad \text{(absolute value of } x_{i+1} - x_i\text{)}$$

$$s_{i-1} + x_i = d_i + s_i, \quad i = 1, \ldots, 12 \quad \text{(inventory balance)}$$

$$\left.\begin{array}{l} x_0 = 0 \\ s_0 = 0 \end{array}\right\} \text{(starting production/inventory)}$$

$$x_i, s_i \geq 0 \quad \text{(non-neg)}$$

$z_i \geq 0$ automatically

Notice we are able to write all constraints compactly by adding the extra variables s_0 and x_0.

How to solve LPs in 2D Graphically

Consider the following LP in 2D:

$$\begin{aligned} \min \quad & 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & 3x_1 + x_2 \geq 4 \\ & 3x_1 + 2x_2 \leq 10 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We demonstrate how we can "plot" this LP, and solve it in that process. Later on, we will extend these techniques to higher dimensions using linear algebra.

The set of all points (x_1, x_2) which satisfy all constraints (including nonnegativity) is called the **feasible region** of the LP.

We first plot the feasible region.

More in the next lecture...

MATH464 – Lecture 6 (01/26/2023)

Today: * graphical solution of LPs in 2D
* Cases of LP

How to solve LPs in 2D Graphically

Consider the following LP in 2D:

$$\begin{aligned} \min \quad & 2x_1+x_2 \\ \text{s.t.} \quad & x_1+x_2 \geq 2 \\ & 3x_1+x_2 \geq 4 \\ & 3x_1+2x_2 \leq 10 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We demonstrate how we can "plot" this LP, and solve it in that process. Later on, we will extend these techniques to higher dimensions using linear algebra.

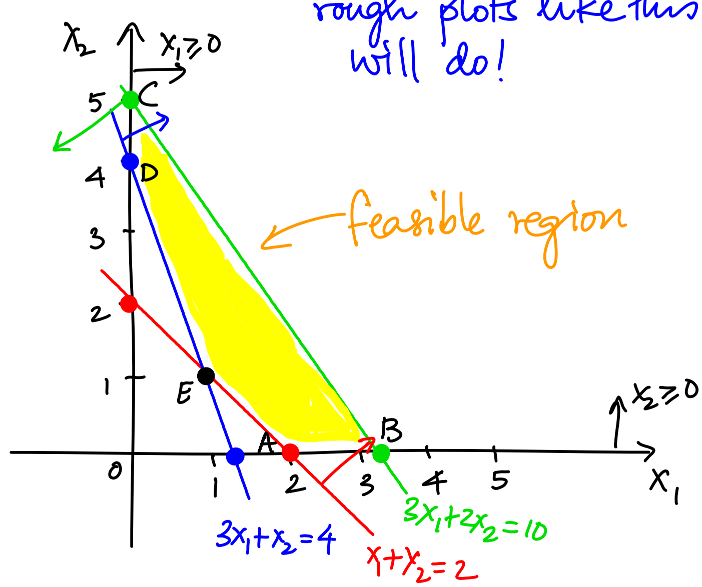
The set of all points (x_1,x_2) which satisfy all constraints (including nonnegativity) is called the **feasible region** of the LP. We first plot the feasible region. To do so, we plot each inequality.

How to plot $x_1+x_2 \geq 2$? We first plot $x_1+x_2=2$. To plot any line, you need two points. The easiest choices are to pick $x_1=0$ and then $x_2=0$ to get the two points, which are $(0,2)$ and $(2,0)$. $x_1+x_2=2$ divides the plane into two half-planes. We need to pick the correct one that is represented by $x_1+x_2 \geq 2$.

To do so, pick any point, say, $(0,0)$ and test it on $x_1+x_2 \geq 2$.

$$0+0 \not\geq 2$$

So $(0,0)$ is on the wrong side. Hence we pick the other side (not containing $(0,0)$), and indicate the choice by drawing the arrow $\rightarrow$ to that side from the line of $x_1+x_2=2$.



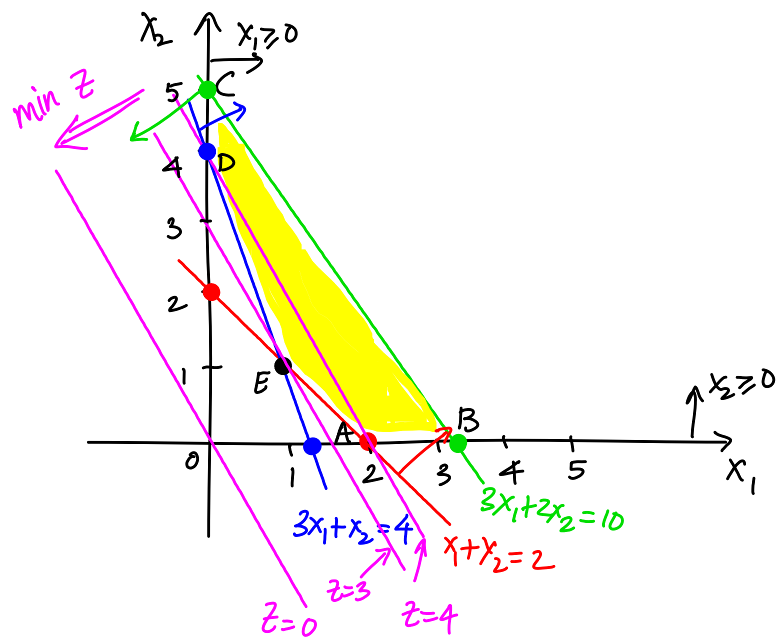
rough plots like this will do!

Once we have plotted all the inequalities, including $x_1 \geq 0$ and $x_2 \geq 0$, the region that is the intersection of all half planes is shaded. This is indeed the feasible region, the polygon ABCDE here.

While it appears E has coordinates $(1,1)$ from the plot, it's better to actually solve for its coordinates, i.e., solve the linear system $\left\{ \begin{array}{l} x_1+x_2=2 \\ 3x_1+x_2=4 \end{array} \right\}$, which indeed gives us $(1,1)$ as the coordinates.

What about the objective function $\min\ 2x_1+x_2$?

Denoting $2x_1+x_2=z$, we plot this line for at least one value of z (may be two), to decide which way to slide it so as to decrease z (recall that we are minimizing z).



Plot the obj. fn for two values of z.

e.g., $z=4$: $2x_1+x_2=4$

We slide the z-line down (min), until we hit $E(1,1)$, where $z=2(1)+(1)=3$. We cannot slide it any further, as we will go out of the feasible region.

So $E(1,1)$, or $x_1=1, x_2=1$ is the optimal solution, giving the optimal objective function value $z^* = 2(1)+(1)=3$.

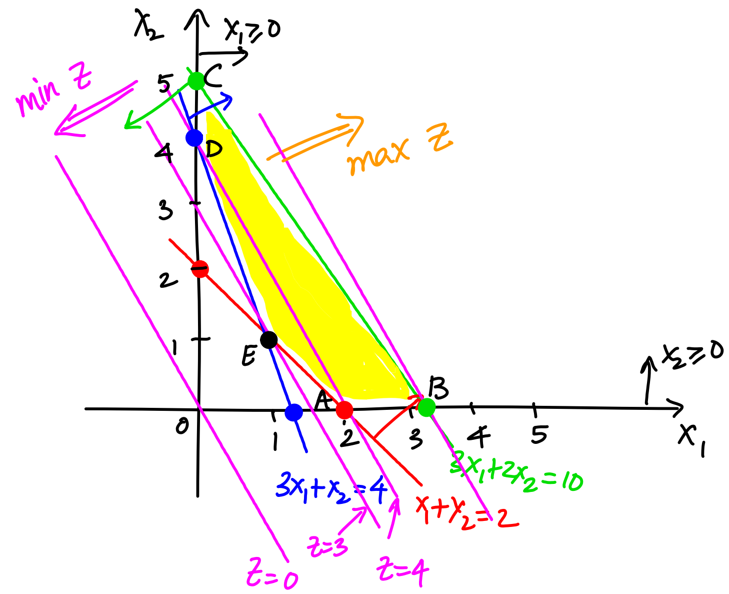
max ~~min~~ $z = 2x_1 + x_2$

s.t. $x_1 + x_2 \geq 2$

$3x_1 + x_2 \geq 4$

$3x_1 + 2x_2 \leq 10$

$x_1, x_2 \geq 0$



Here, E(1,1) is the unique optimal solution. To make sure, we check z value at A(2,0) and D(0,4): $z(A) = 2(2)+1(0) = 4$ and $z(D) = 2(0)+1(4) = 4$, both are $> z(E) = 3$.

In the case of a system of linear equations $A\bar{x}=\bar{b}$, we have three possibilities— the system has a unique solution, it has infinitely many solutions, or it is inconsistent.
We get corresponding cases for LP, but get one more case in addition.

This LP is an example of **Case I** (or **Type I**), where the LP has a **unique optimal solution**.

Consider a small variation now:

For $\left\{ \begin{array}{l} \max\ z = 2x_1 + x_2 \\ \text{s.t. same constraints} \end{array} \right\}$ $B(10/3, 0)$ is the unique optimal solution.

This max-LP also is of Case I.

$z(B) = 2(10/3) + 1(0) = \frac{20}{3}$.

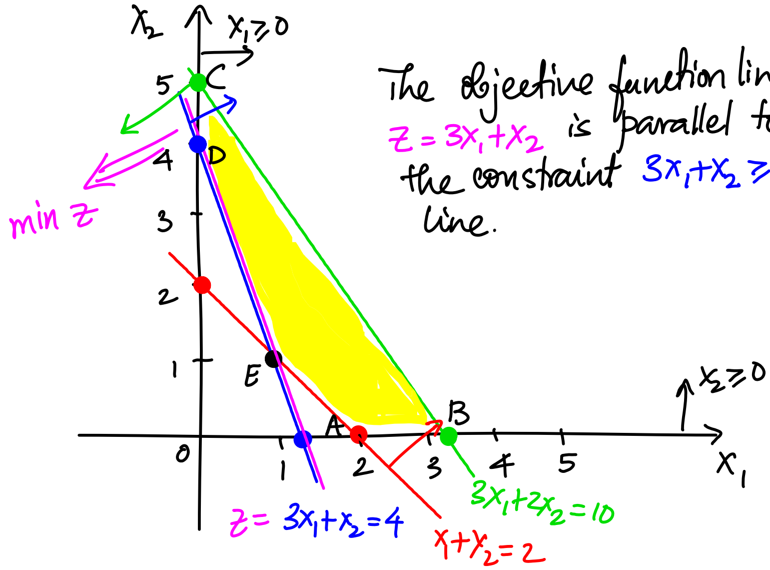
$z(C) = 2(0) + 1(5) = 5 < \frac{20}{3}$.

Consider a slightly different LP:

$$\min\ Z = 3x_1 + x_2$$

s.t.

$$x_1 + x_2 \geq 2$$
$$3x_1 + x_2 \geq 4$$
$$3x_1 + 2x_2 \leq 10$$
$$x_1, x_2 \geq 0$$



The objective function line $z = 3x_1 + x_2$ is parallel to the constraint $3x_1 + x_2 \geq 4$ line.

Here, the z-line hits the entire line segment $\overline{DE}$. Every point in the line segment is an optimal solution. So the LP has **alternative optimal solutions**, and belongs to **Case II**.

Recall: parametric vector form of solutions to $A\bar{x} = \bar{b}$.
For instance, if there were two free variables, we could write all solutions in the form $\bar{x} = \bar{x}_0 + p\bar{r} + q\bar{s}$, where $p, q \in \mathbb{R}$, and $A\bar{x}_0 = \bar{b}$, i.e., $\bar{x}_0$ is a particular solution, and $A\bar{r} = \bar{0}$, $A\bar{s} = \bar{0}$, i.e., both $\bar{r}$ and $\bar{s}$ belong to $\mathrm{Nul}(A)$.

Notice that $E(1,1)$ and $D(0,4)$ both give $z^* = 3x_1 + x_2 = 4$.

We can describe all optimal solutions to this LP as

$$\bar{x} = \lambda \underbrace{\begin{bmatrix} 0 \\ 4 \end{bmatrix}}_{D} + (1-\lambda) \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{E}, \quad \lambda \in [0,1] \quad \text{i.e.,} \quad \bar{x} = \begin{bmatrix} 1-\lambda \\ 3\lambda+1 \end{bmatrix}, \quad \lambda \in [0,1].$$

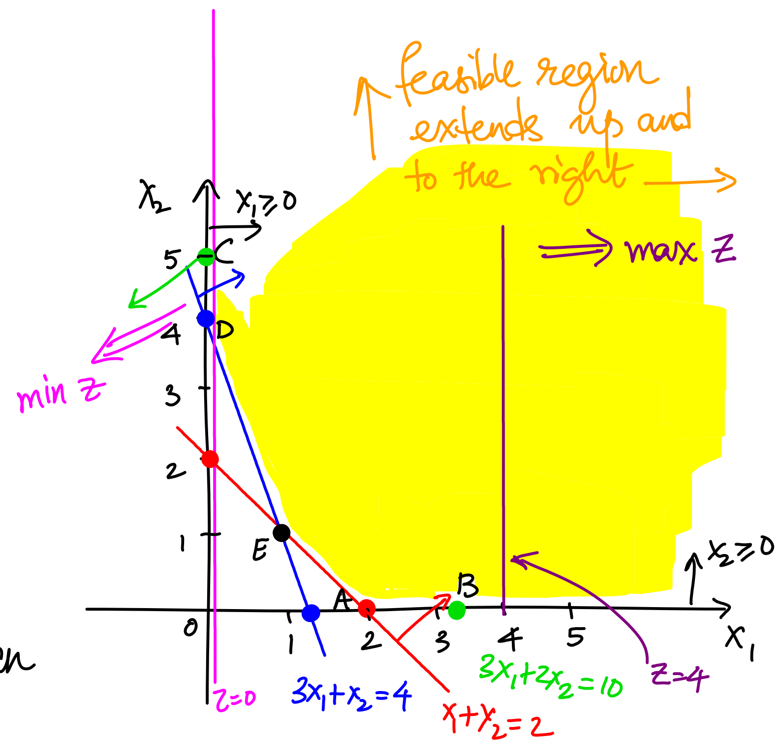
Indeed, $\bar{c}^T\bar{x} = \begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} 1-\lambda \\ 3\lambda+1 \end{bmatrix} = 3 - 3\lambda + 3\lambda + 1 = 4.$

$\bar{c} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ is the objective function coefficient vector

In higher dimensions, we could get more that two "vertices" (such as D & E here), and the entire "face" between them as the set of optimal solutions.

We consider another LP of Case II below.

$$\min \quad x_1$$
$$\text{s.t.} \quad x_1+x_2 \geq 2$$
$$3x_1+x_2 \geq 4$$
$$x_1, x_2 \geq 0$$



Here, any point on the vertical axis at or above D(0,4) is an optimal solution.

Here, the optimal solution set is the ray going vertically up from D(0,4) (or the half-open line from D(0,4) up). While we again have infinitely many optimal solutions here, notice that the optimal solution set is **unbounded**, unlike $\overline{DE}$ in the previous case.

We can describe all optimal solutions here as

$$\bar{x} = \underbrace{\begin{bmatrix}0\\4\end{bmatrix}}_{D} + \lambda \begin{bmatrix}0\\1\end{bmatrix}, \lambda \in \mathbb{R}, \lambda \geq 0.$$

$\rightarrow$ the unit vector pointing up along the vertical axis

We now consider a slight modification of the above LP.

$$\max \quad x_1$$
$$\text{s.t.} \quad x_1+x_2 \geq 2$$
$$3x_1+x_2 \geq 4$$
$$x_1, x_2 \geq 0$$

$z=x_1$ line could be pushed to the right without limit, i.e., there is no finite optimal z-value. We say the LP is **unbounded**, and it belongs to **Case III**.

This is the extra case which does not have a corresponding case in $A\bar{x}=\bar{b}$ (system of linear equations).

Notice the difference between unbounded feasible region and unbounded LP. If the feasible region is bounded, the LP cannot be unbounded as the z-line cannot be moved in the improving direction without limits. On the other hand, the feasible region could be unbounded but the LP could still have an optimal solution, as seen above (variation of Case II).

4. Case IV LP (infeasible LPs)

Consider a slightly different LP:

$$\min\ Z = 3x_1 + x_2$$

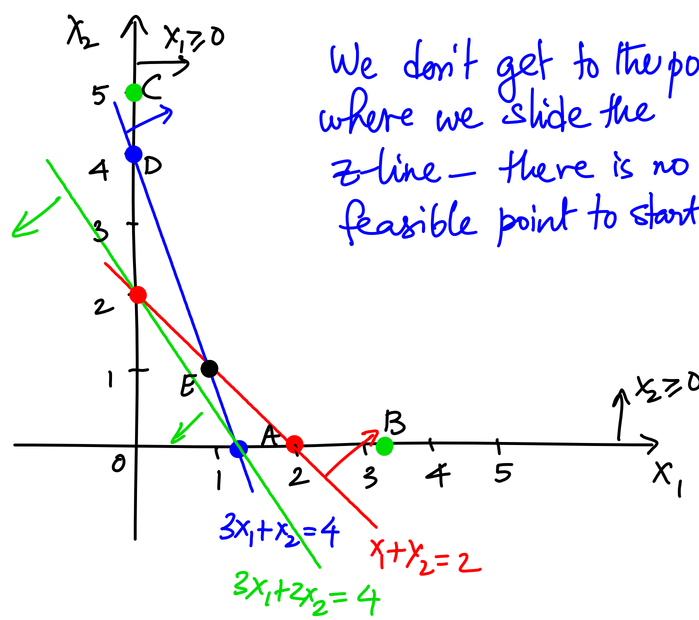
s.t.

$$x_1 + x_2 \geq 2$$

$$3x_1 + x_2 \geq 4$$

$$3x_1 + 2x_2 \leq 4$$

$$x_1, x_2 \geq 0$$



We don't get to the point where we slide the z-line — there is no feasible point to start!

There are no feasible solutions, i.e, there are no points satisfying all constraints. So the LP is **infeasible**. So its feasible region is empty.

In practice, if an LP formulation comes out to be infeasible, it could indicate that, for instance, we cannot meet all demand using the resources available. Or that the cost for a particular necessary project will not fit within the budget.

MATH464 – Lecture 7 (01/31/2023)

Today: * Results on polyhedra

Background in Linear Algebra and Geometry

We will present definitions and results related to polyhedra and convexity. The goal is to generalize the graphical solution of LP in 2D to high dimensions.

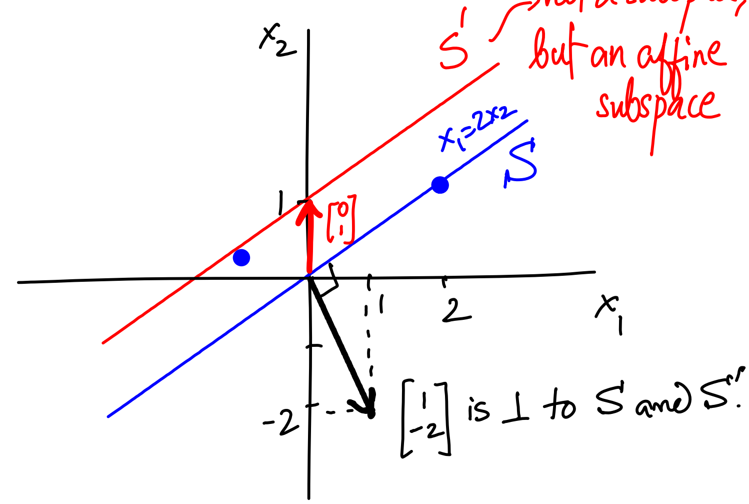
Subspaces: A set $S \subseteq \mathbb{R}^n$ is a subspace if $\forall\, \bar{x}, \bar{y} \in S$, $a\bar{x} + b\bar{y} \in S$ for all $a, b \in \mathbb{R}$.

In particular, $\bar{0} \in S$ (for $a=b=0$). → we usually mention the inclusion of $\bar{0}$ separately (to define a subspace).

zero vector

e.g. in 2D, any line passing through the origin is a subspace of $\mathbb{R}^2$

$x_1 = 2x_2$ or $x_1 - 2x_2 = 0$, i.e., $\begin{bmatrix} 1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$.



All points on S can be described as $\lambda\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ for $\lambda \in \mathbb{R}$.

So, $\left\{\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right\}$ is a basis for S, and $\dim S = 1$ (dimension of S).

If S is strictly smaller than $\mathbb{R}^n$, we say that S is a proper subspace of $\mathbb{R}^n$.

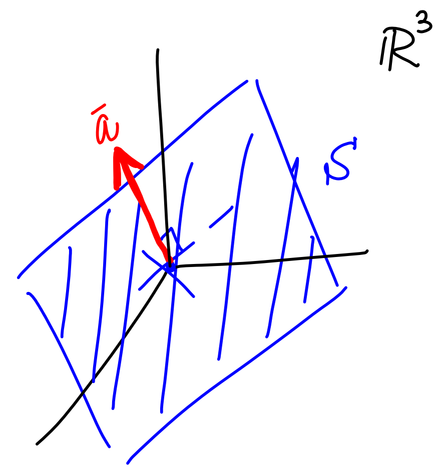
In $\mathbb{R}^3$, planes passing through the origin are proper subspaces.

Affine subspace: Given a subspace S' of $\mathbb{R}^n$, the set $S_0 = \bar{x}_0 + S' = \{\bar{x}_0 + \bar{x} \mid \bar{x} \in S'\}$ is an affine subspace for $\bar{x}_0 \notin S'$.

Each proper subspace and affine subspace in $\mathbb{R}^2$ has a vector orthogonal to it. In higher dimensions, if S' is an m-dimensional subspace of $\mathbb{R}^n$ with $m < n$, there will be $n-m$ linearly independent (LI) vectors orthogonal to S'.

In the 2D example above, $\bar{a} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$ is orthogonal to both $S = \{\bar{x} \mid \bar{x} = \lambda \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \lambda \in \mathbb{R}\}$ and $S' = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + S$.

Illustration in $\mathbb{R}^3$ for the case when $\dim S' = 2$.



Back to feasible region of LP...

We now introduce polyhedra, which are generalizations of "nice," i.e., convex polygons in 2D to higher dimensions. The feasible region of an LP is a polyhedron.

≥ 0 are included

Def A **polyhedron** is $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \geq \bar{b}\}$, where $A_{m \times n}$, $\bar{b}_{m \times 1}$ have real entries.

(m, n are finite here)

$\bar{a}_i^T \bar{x} \geq b_i$ for $i = 1, \ldots, m$

m linear inequalities

In words, P is a set of points satisfying a finite set of linear inequalities. If there are non-negativity constraints involved, they are included as part of the main set of constraints $A\bar{x} \geq \bar{b}$.

If there are equations of the form $\bar{a}_i^T\bar{x} = b_i$, we could split each of them into a pair of inequalilities of the form $\bar{a}_i^T\bar{x} \geq b_i$ and $\bar{a}_i^T\bar{x} \leq b_i$, and rewrite the latter inequalities as $-\bar{a}_i^T\bar{x} \geq -b_i$.

Hyperplane: $\{\bar{x} \in \mathbb{R}^n \mid \bar{a}^T\bar{x} = b\}$, $\bar{a} \in \mathbb{R}^n$, $b \in \mathbb{R}$

e.g., $ax+by+cz=d$ is a plane in 3D

Half-space: $\{\bar{x} \in \mathbb{R}^n \mid \bar{a}^T\bar{x} \geq b\}$.

Result The feasible region of an LP is a polyhedron.

Result A polyhedron is the intersection of half-planes.

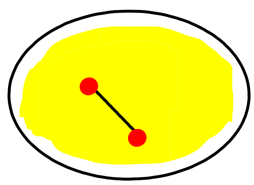
Recall how we drew the feasible region in 2D. After selecting the correct side of each equation (indicated by the arrows $\rightarrow$ or $\leftarrow$), we pick the region common to all the selected half-spaces.

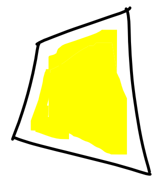
We previosly defined convex sets:

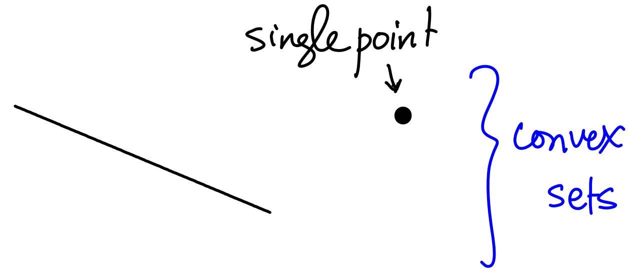
Def A set $S \subseteq \mathbb{R}^n$ is a **convex set** if for all $\bar{x}, \bar{y} \in S$ and $\lambda \in [0,1]$, $\underline{\lambda\bar{x} + (1-\lambda)\bar{y}} \in S$.

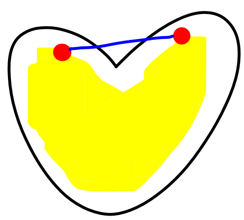
$\longrightarrow$ convex combination of $\bar{x}$ and $\bar{y}$

In words, the line segment connecting any two points in the set lies entirely within the set.

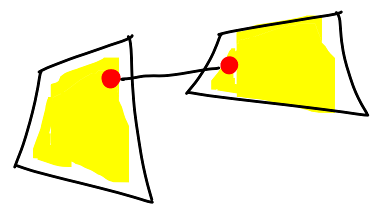




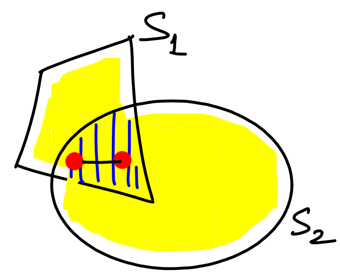




not convex !



union of convex sets need not be convex



But intersections of convex sets are convex

BT-ILO Theorem 2.1 A polyhedron is a convex set.

Proof We present the proof by showing two results.

(a) Intersection of convex sets is convex.

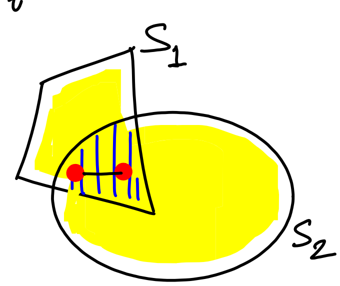
Let $S_i,\ i \in I$ be convex, and $S = \bigcap_{i\in I} S_i$, the intersection of S_i.

Let $\bar{x}, \bar{y} \in S \Rightarrow \bar{x}, \bar{y} \in S_i\ \forall i \in I$.

Since S_i is convex, $\lambda \bar{x} + (1-\lambda)\bar{y} \in S_i$ for $\lambda \in [0,1]$ $\forall i \in I$.

$\Rightarrow \lambda \bar{x} + (1-\lambda)\bar{y} \in S$ (as $S = \bigcap_{i\in I} S_i$)

$\Rightarrow S$ is convex.



I need not be finite here

(b) Half space $H = \{\bar{x} \in \mathbb{R}^n \mid \bar{a}^T\bar{x} \geqslant b\}$ is convex.

Let $\bar{x}, \bar{y} \in H \Rightarrow \underline{\bar{a}^T\bar{x} \geqslant b}$ and $\underline{\bar{a}^T\bar{y} \geqslant b}$

$\Rightarrow \bar{a}^T(\lambda\bar{x} + (1-\lambda)\bar{y}) = \lambda\, \bar{a}^T\bar{x} + (1-\lambda)\, \bar{a}^T\bar{y}$ for $\lambda \in [0,1]$

$\geqslant \lambda\, b + (1-\lambda)\, b = b$

$\Rightarrow \lambda\bar{x} + (1-\lambda)\bar{y} \in H$, i.e., H is convex.

Polyhedron P is the intersection of m half-spaces $H_i = \{\bar{x} \in \mathbb{R}^n \mid \bar{a}_i^T \bar{x} \geqslant b_i\}$.
Then by (a) and (b) above, we get that P is convex $\square$

Alternatively, we could present a more direct proof. Let $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \geqslant \bar{b}\}$, and let $\bar{x}, \bar{y} \in P$. Then $A\bar{x} \geqslant \bar{b}$ and $A\bar{y} \geqslant \bar{b}$. Hence

$$A(\lambda\bar{x} + (1-\lambda)\bar{y}) \geqslant \lambda\bar{b} + (1-\lambda)\bar{b} = \bar{b} \Rightarrow \lambda\bar{x} + (1-\lambda)\bar{y} \in P \text{ as well.}$$

Some more definitions

Def A set $S \subset \mathbb{R}^n$ is **bounded** if $\max\limits_{i=1,\dots,n} \{|x_i| \mid \bar{x} \in S\} \leq K$ for some finite non-negative number K.

A bounded polyhedron is called a **polytope**.

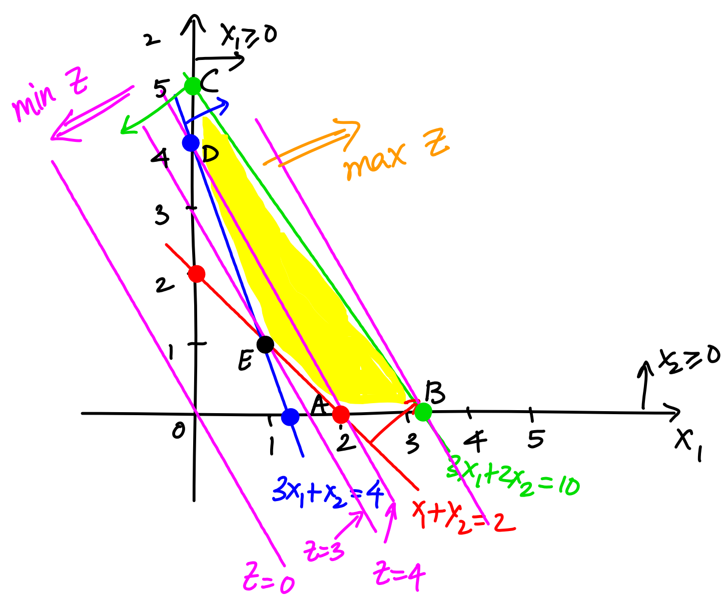
~~min~~ max $z = 2x_1 + x_2$

s.t. $x_1 + x_2 \geqslant 2$

$3x_1 + x_2 \geqslant 4$

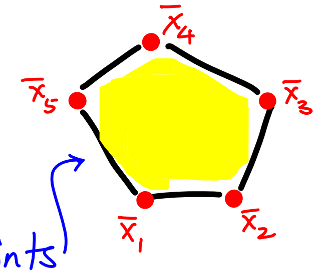
$3x_1 + 2x_2 \leq 10$

$x_1, x_2 \geqslant 0$



The feasible region ABCDE is bounded, e.g., $K=6$ will work.
But if we remove $3x_1+2x_2 \leq 10$, the feasible region is no longer bounded.

Recall A convex combination of $\bar{x}_1, \dots, \bar{x}_m \in \mathbb{R}^n$ is
$$\bar{x} = \sum_{i=1}^{m} \lambda_i \bar{x}_i, \quad \lambda_i \geq 0 \; \forall i, \quad \sum_{i=1}^{m} \lambda_i = 1.$$



strap an elastic band around all points

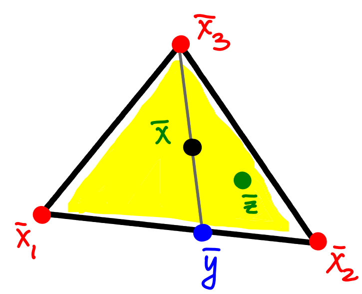
Def The **convex hull** of $\bar{x}_1, \dots, \bar{x}_m = \{\bar{x} \in \mathbb{R}^n \mid \bar{x} = \sum_{i=1}^{m} \lambda_i \bar{x}_i, \; \lambda_i \geq 0, \; \sum_i \lambda_i = 1\}$.
denoted $\text{conv}(\bar{x}_1, \dots, \bar{x}_m)$.

e.g., The convex hull of A, B, C, D, E is the feasible region of the LP.

BT-ILO Theorem 2.1(d) The convex hull of $\bar{x}_1, \dots, \bar{x}_m \in \mathbb{R}^n$ is a convex set.

Read proof in book! Intuitive idea is presented here.

Could use an "inductive" argument. Consider $\text{conv}(\{\bar{x}_1, \bar{x}_2\})$.
$\bar{y} = \lambda \bar{x}_1 + (1-\lambda)\bar{x}_2$ for some $\lambda \in [0,1]$. Any point in $\text{conv}(\{\bar{x}_1, \bar{x}_2, \bar{x}_3\})$ can be written as
$$\bar{x} = \mu \bar{y} + (1-\mu)\bar{x}_3 \text{ for } \mu \in [0,1].$$



Now consider another point $\bar{z} \in \text{conv}(\{\bar{x}_1, \bar{x}_2, \bar{x}_3\})$. We can show that $\eta \bar{x} + (1-\eta)\bar{z} \in \text{conv}(\{\bar{x}_1, \bar{x}_2, \bar{x}_3\})$.

Recall how E(1,1), the optimal solution in the previous LP, was a vertex or corner point of the feasible region ABCDE. Indeed, we can generalize this observation — we can look of "vertices" of feasible regions for candidate optimal solutions. We first formalize the notion of a vertex — both geometrically and algebraically.

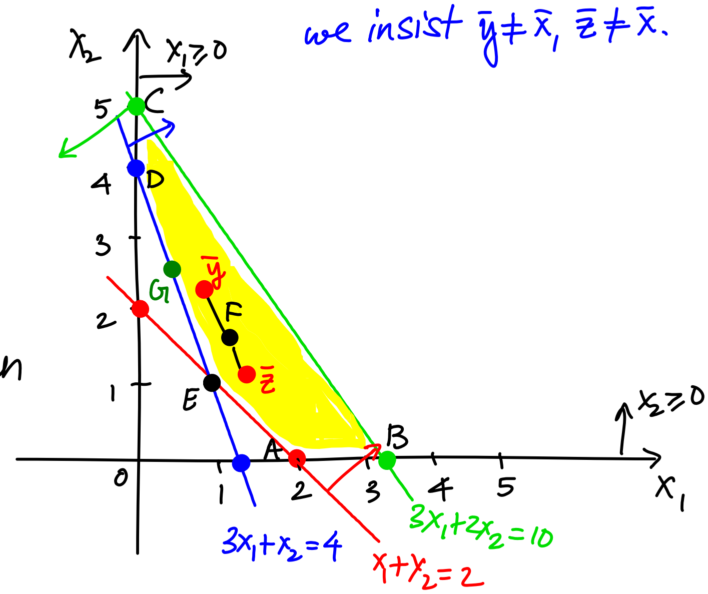
Vertices, Extreme points, and Basic Feasible Solutions

We are given a polyhedron $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \geq \bar{b}\}$.

Def $\bar{x} \in P$ is an **extreme point** of P if there do not exist $\bar{y}, \bar{z} \in P$, $\bar{y} \neq \bar{x}$, $\bar{z} \neq \bar{x}$, such that $\bar{x} = \lambda\bar{y} + (1-\lambda)\bar{z}$ for some $\lambda \in [0,1]$.

λ could be 0 or 1, but we insist $\bar{y} \neq \bar{x}$, $\bar{z} \neq \bar{x}$.

In words, you cannot write an extreme point as a convex combination of two other distinct points.



E is an extreme point, but F and G are not. For instance, G can be written as a convex combination of D and E.

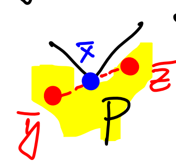
MATH 464 – Lecture 8 (02/02/2023)

Today: * vertex, basic feasible solution (bfs)
* extreme point $\Leftrightarrow$ vertex $\Leftrightarrow$ bfs

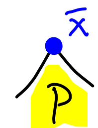
---

Recall: $\bar{x} \in P$ is an **extreme point** if $\nexists\, \bar{y}, \bar{z} \in P$, $\bar{y} \neq \bar{x}$, $\bar{z} \neq \bar{x}$, s.t. $\bar{x} = \lambda \bar{y} + (1-\lambda)\bar{z}$ for some $\lambda \in [0,1]$.

P being convex is crucial here.

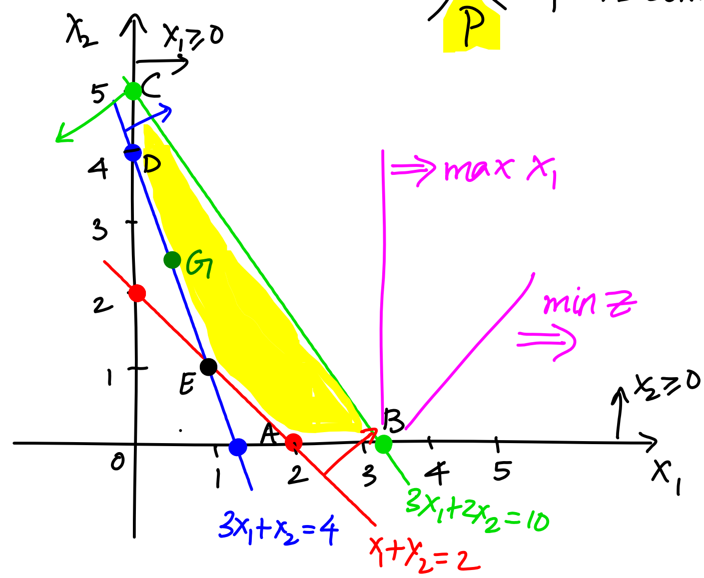


P not convex



P is convex

$$\begin{aligned} \min \quad & 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & 3x_1 + x_2 \geq 4 \\ & 3x_1 + 2x_2 \leq 10 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Def $\bar{x} \in P$ is a **vertex** of P if there is a vector $\bar{c} \in \mathbb{R}^n$ such that $\bar{c}^T\bar{x} < \bar{c}^T\bar{y}$ $\forall\, \bar{y} \in P$, $\bar{y} \neq \bar{x}$.

So, $\bar{x}$ is the unique optimal solution of some LP with P as the feasible region.

$\max x_1 \equiv \min -x_1$

E is a vertex, $\bar{c} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ works. Similarly B is a vertex, and $\bar{c} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ works for B. But G is not a vertex – no $\bar{c}$ has G as the unique optimal solution. Recall that $\bar{c} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ has $\overline{DE}$ (the entire line segment) as the set of optimal solutions.

There could be many $\bar{c}$ for which B is the unique optimal solution to $\min \bar{c}^T\bar{x}$.

The above two definitions (extreme point and vertex) are geometric, and hence intuitive. But we need an equivalent definition that is algebraic, so that we could do computations.

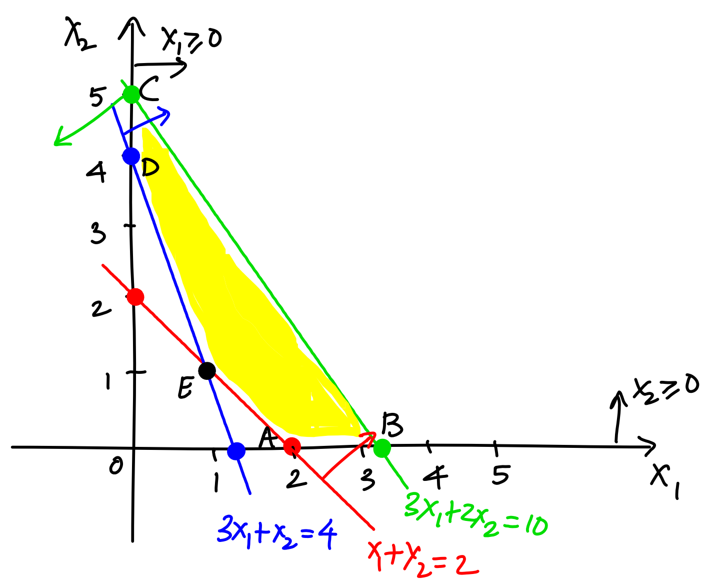
We need a few more concepts first.

Active/binding Constraint

Def A constraint $\bar{a}_i^T \bar{x} \geq b_i$ is called **binding** or **active** at $\bar{x}^* \in P$ if $\bar{a}_i^T \bar{x}^* = b_i$, i.e., the constraint is satisfied as an equation.

Note: Equality constraints are always active (at all $\bar{x} \in P$).

$$\begin{aligned} \min \quad & 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & 3x_1 + x_2 \geq 4 \\ & 3x_1 + 2x_2 \leq 10 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Thus, $3x_1+2x_2 \leq 10$ is active at B, but not active at A or E.

Notice that at each vertex A, B, C, D, and E, two constraints are active. For instance, at B, $3x_1+2x_2 \leq 10$ and $x_2 \geq 0$ are binding. At E, $x_1+x_2 \geq 2$ and $3x_1+x_2 \geq 4$ are active. At D, $3x_1+x_2 \geq 4$ and $x_1 \geq 0$ are active.

We formalize this intuition to higher dimensions. If there are n constraints active at $\bar{x}^*$, then $\bar{x}^*$ is a solution to a system of those n linear equations. If the vectors $\bar{a}_i$ defining these n constraints are LI, then $\bar{x}^*$ is the unique solution to this linear system. We formalize these concepts in the following definition.

Def Let P be a polyhedron with equality and inequality constraints.

(a) $\bar{x}^* \in \mathbb{R}^n$ is a **basic solution** of P if

$\bar{x}^*$ need not be feasible here!

(i) all equality constraints are active; and

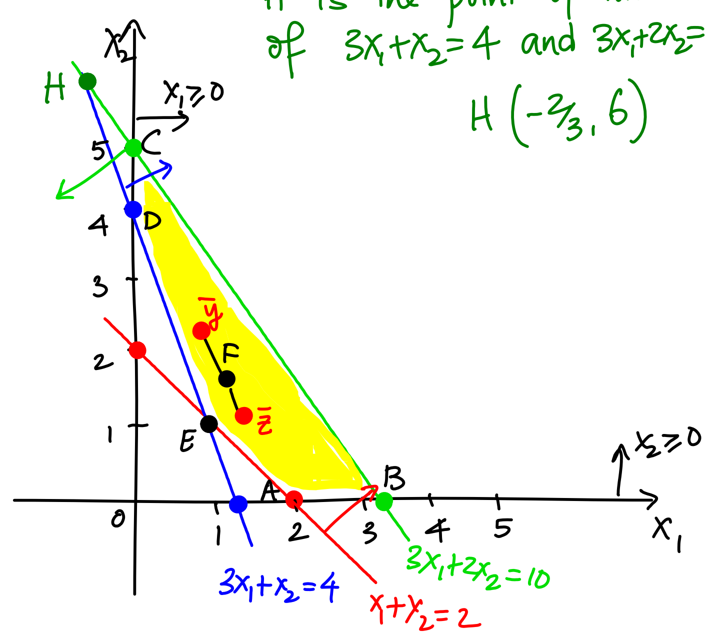
(ii) out of all constraints active at $\bar{x}^*$, there are n of them which are LI.

If the constraints are $\bar{a}_i^T \bar{x} = b_i$ for $i \in I$, then $\{\bar{a}_i\}_{i \in I}$ is LI.

(b) If $\bar{x}^*$ is a basic solution that satisfies all constraints of P, then it is a **basic feasible solution** (**bfs**).

H is a basic solution, but not a bfs

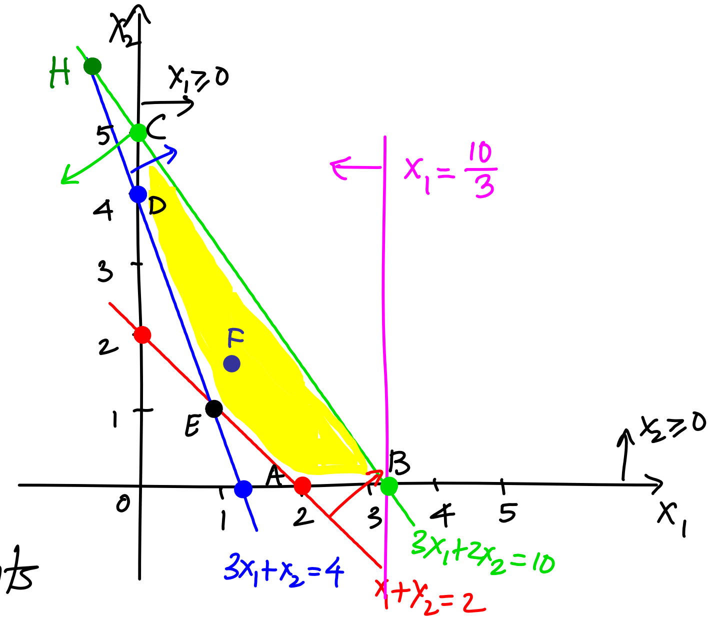
A, B, C, D, E are bfs's.



H is the point of intersection of $3x_1+x_2=4$ and $3x_1+2x_2=10$

$H\left(-\frac{2}{3}, 6\right)$

Notice that F is not a basic solution, even though it is feasible, since none of the constraints are active at F.



Now we add another constraint to the original set of 3 constraints (and nonnegativity)

$$
\begin{aligned}
x_1+x_2 &\geq 2\\
3x_1+x_2 &\geq 4\\
3x_1+2x_2 &\leq 10\\
x_1 \quad &\leq \frac{10}{3}\\
x_1, x_2 &\geq 0
\end{aligned}
$$

Now, three constraints are active at B: $3x_1+2x_2 \leq 10$, $x_1 \leq \frac{10}{3}$, $x_2 \geq 0$. Any subset of two of these constraints is LI. Equivalently, with $\underset{3x_1+2x_2\leq 10}{\bar{a}_3 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}}$, $\underset{x_1 \leq \frac{10}{3}}{\bar{a}_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}}$, $\underset{x_2 \geq 0}{\bar{a}_6 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}}$,

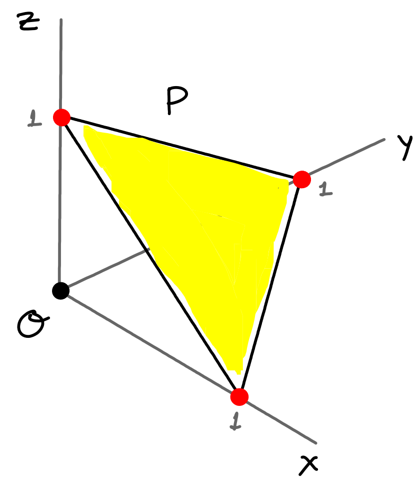
any two of $\{\bar{a}_3, \bar{a}_4, \bar{a}_6\}$ are LI. So B is still a bfs

$\bar{a}_i$ is the coefficient vector of the i^{th} constraint: $\bar{a}_i^T \bar{x} \geq b_i$.

$x_1 \geq 0$, $x_2 \geq 0$ are the 5^{th} and 6^{th} constraints, after adding $x_1 \leq 10/3$ as the 4^{th} constraint.

Recall definition of bfs; in particular to requirement (a) i.

$$P = \{ \bar{x} \in \mathbb{R}^3 \mid x_1 + x_2 + x_3 = 1,\ x_j \geq 0\ \forall j \}.$$



Note that O (the origin) has 3 LI constraints active ($x_j \geq 0\ \forall j$).

But O is not a basic solution, as it does not satisfy the equality constraint.

But if we write

$$P = \left\{ \bar{x} \in \mathbb{R}^3 \,\middle|\, \begin{array}{l} x_1 + x_2 + x_3 \leq 1, \\ x_1 + x_2 + x_3 \geq 1 \end{array} \ x_j \geq 0\ \forall j \right\},$$

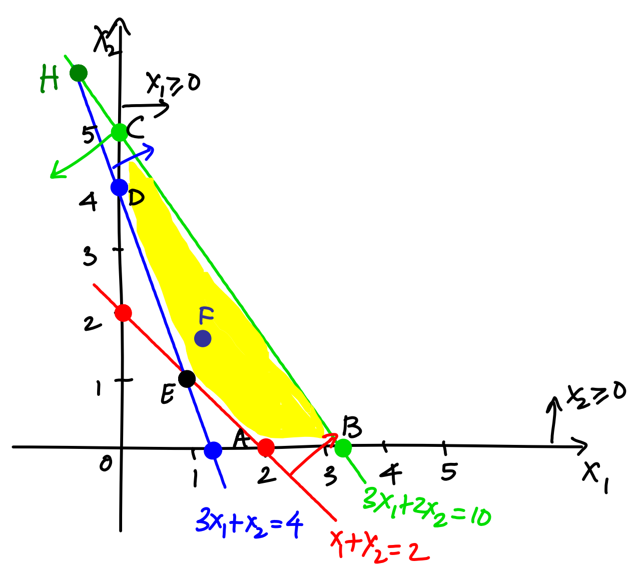
O is a basic solution now!

Hence, the way we present inequalities matter!

But there is no such confusion for bfs's!

Back to our example:

At $D(0,4)$, $3x_1 + x_2 \geq 4$ and $x_1 \geq 0$ are active, and are LI.



Theorem 2.3 (BT-ILO) Extreme point $\Longleftrightarrow$ vertex $\Longleftrightarrow$ bfs.

see BT-ILO!

Or, the definitions are equivalent.

Proof 1. (vertex $\Rightarrow$ extreme point)

The proof is presented for two implications here.

Let $\bar{x}^*$ be a vertex of P. Hence there exists $\bar{c} \in \mathbb{R}^n$ such that $\bar{c}^T \bar{x}^* < \bar{c}^T \bar{y} \;\; \forall \, \bar{y} \in P, \; \bar{y} \neq \bar{x}^*$.

Let $\bar{y}, \bar{z} \in P$, $\bar{y} \neq \bar{x}^*$, $\bar{z} \neq \bar{x}^*$, and $\lambda \in (0,1)$.

we do not include the end points, as we want $\bar{y}, \bar{z} \neq \bar{x}^*$.

So we get $\bar{c}^T \bar{x}^* < \bar{c}^T (\underbrace{\lambda \bar{y} + (1-\lambda)\bar{z}}_{\text{is } \neq \bar{x}^*})$ for any $\lambda \in (0,1)$.

$\Rightarrow \; \bar{x}^* \neq \lambda \bar{y} + (1-\lambda)\bar{z}$, or we cannot write $\bar{x}^*$ as a convex combination of $\bar{y}$ and $\bar{z}$. So, $\bar{x}^*$ is an extreme point.

2. (extreme point $\Rightarrow$ bfs)

We prove, equivalently, that (not bfs) $\Rightarrow$ (not extreme point).

Contrapositive argument: $(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A)$ for logical statements A, B

"negation" or "not"

"equivalent"

Let $\bar{x}^* \in P$ be not a bfs. Hence among all the active constraints at $\bar{x}^*$, there do not exist a set of n LI constraints.

Let $\bar{a}_i^T \bar{x}^* = b_i$, $i \in I$ be the set of constraints active at $\bar{x}^*$.

Hence, $\{\bar{a}_i \in \mathbb{R}^n, i \in I\}$ are not LI.

index set

Hence, there exists some $\bar{d} \in \mathbb{R}^n$, $\bar{d} \neq \bar{0}$, such that $\bar{a}_i^T \bar{d} = 0 \; \forall i \in I$.

Recall from linear algebra that if the columns of $A = [\bar{a}_1 \cdots \bar{a}_k]$ are not LI, then the homogeneous system $A\bar{x} = \bar{0}$ has a non-trivial solution.

Let $\bar{y} = \bar{x}^* + \epsilon \bar{d}$ and $\bar{z} = \bar{x}^* - \epsilon \bar{d}$ for $\epsilon > 0$, but small.

We want to show that $\bar{y}, \bar{z} \in P$ for ϵ small enough. Since $\bar{x}^* = \frac{1}{2}(\bar{y} + \bar{z})$, that will certify that $\bar{x}^*$ is <u>not</u> an extreme point.

We get $\bar{a}_i^T \bar{y} = b_i \; \forall i \in I$ as $\bar{a}_i^T (\bar{x}^* + \epsilon \bar{d}) = b_i + \epsilon 0 = b_i$.

We'll finish the proof in the next lecture...

MATH464 – Lecture 9 (02/07/2023)

Today: * bfs ⟹ extreme point (proof)
* finding all basic solutions

Proof of Theorem 2.3 (contd..) 2. Extreme pt ⟹ bfs

We prove, equivalently, that (not bfs) ⟹ (not extreme point).

contrapositive argument: $(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A)$

for logical statements A, B

"equivalent"

"negation" or "not"

Let $\bar{x}^* \in P$ be not a bfs. Hence among all the active constraints at $\bar{x}^*$, there do not exist a set of n LI constraints.

Let $\bar{a}_i^T \bar{x}^* = b_i$, $i \in I$ be the set of constraints active at $\bar{x}^*$.

Hence, $\{\bar{a}_i \in \mathbb{R}^n, i \in I\}$ are not LI.

index set

Hence, there exists some $\bar{d} \in \mathbb{R}^n$, $\bar{d} \neq \bar{0}$, such that $\bar{a}_i^T \bar{d} = 0 \ \forall i \in I$.

Recall from linear algebra that if the columns of $A = [\bar{a}_1 \cdots \bar{a}_k]$ are not LI, then the homogeneous system $A\bar{x} = \bar{0}$ has a non-trivial solution.

Let $\bar{y} = \bar{x}^* + \epsilon \bar{d}$ and $\bar{z} = \bar{x}^* - \epsilon \bar{d}$ for $\epsilon > 0$, but small.

We want to show that $\bar{y}, \bar{z} \in P$ for ϵ small enough.

We get $\bar{a}_i^T \bar{y} = b_i \ \forall i \in I$ as $\bar{a}_i^T(\bar{x}^* + \epsilon \bar{d}) = b_i + \epsilon 0 = b_i$.

If $i \notin I$, $\bar{a}_i^T \bar{x}^* > b_i$. (We take $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \geqslant \bar{b}\}$

or $\bar{a}_i^T \bar{x} \geqslant b_i, i=1,\dots,m$)

constraint i is not active

We also get $\bar{a}_i^T \bar{y} = \bar{a}_i^T(\bar{x}^* + \epsilon \bar{d}) > b_i + \epsilon \underbrace{\bar{a}_i^T \bar{d}}_{\neq 0 \text{ (as } i \notin I)} > b_i$ for ϵ small enough.

In particular, we can choose $\epsilon < \max\limits_{i \in I} \left\{ \dfrac{(\bar{a}_i^T \bar{x}^* - b_i)}{|\bar{a}_i^T \bar{d}|} \right\}$.

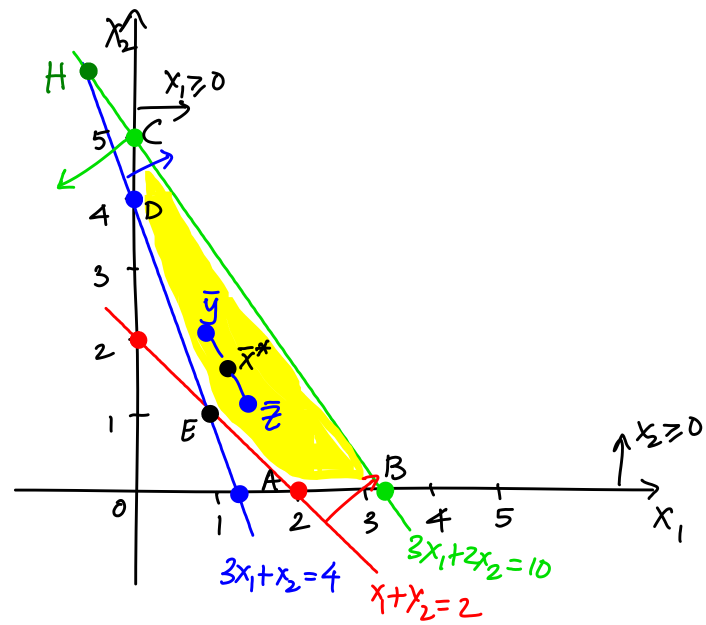
Thus, $\bar{a}_i^T \bar{y} = b_i \ \forall i \in I$, and $\bar{a}_i^T \bar{y} > b_i \ \forall i \notin I$. So $\bar{y} \in P$.

Ultimately, we choose the smaller of the two ϵ values (for $\bar{y}, \bar{z}$).

Similarly, $\bar{z} \in P$. But $\bar{x}^* = \frac{1}{2}(\bar{y} + \bar{z})$, i.e., $\bar{x}^*$ is a convex combination of $\bar{y}, \bar{z} \in P$, $\bar{y} \neq \bar{x}^*, \bar{z} \neq \bar{x}^*$.

So $\bar{x}^*$ is not an extreme point.

Intuitively, if $\bar{x}^*$ is feasible but is "in the interior" of P, we can find two other points close enough to $\bar{x}^*$, but still in P, such that $\bar{x}^*$ is the midpoint of the line segment connecting those two points.



3. (bfs $\Rightarrow$ vertex)

see the book (BT-ILO for details).

$\square$

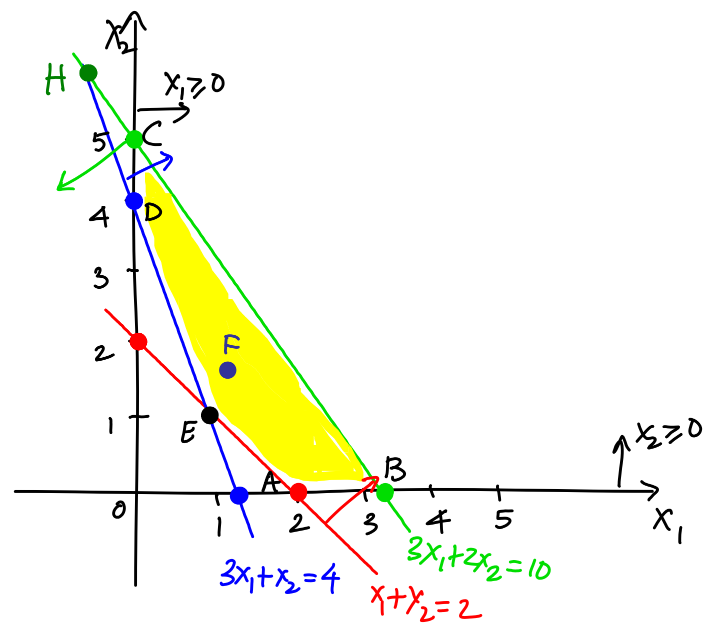
We are almost ready to describe how to go about finding the basic solutions as well as bfs's. But we introduce one more concept first.

Adjacent Basic Solutions

Def Two basic solutions are **adjacent** if there are the same $(n-1)$ LI constraints that are active at both of them. If two adjacent basic solutions are both feasible, then they are **adjacent bfs's**.

e.g., A and B are adjacent bfs's, as $x_2 \geq 0$ is active at both of them (here, $n-1=1$).

H and C are adjacent basic solutions, as $3x_1+2x_2 \leq 10$ is active at both H and C.



Since H is not feasible, H and C are not adjacent bfs's.

As we will see later, we will move from one bfs to an adjacent bfs that improves the objective function value. We try to repeat this step until we cannot go to an adjacent bfs that improves the objective function value any more. Then we have an optimal bfs.

We describe the method to identify basic (feasible) solutions for polyhedra in standard form. Recall that the LP $\begin{Bmatrix} \min \bar{c}^T\bar{x} \\ \text{s.t. } A\bar{x}=\bar{b} \\ \bar{x}\geq\bar{0} \end{Bmatrix}$ is in standard form.

Polyhedra in standard form

Def $P = \{\bar{x}\in\mathbb{R}^n \mid A\bar{x}=\bar{b}, \bar{x}\geq\bar{0}\}$, $A\in\mathbb{R}^{m\times n}$, $\bar{b}\in\mathbb{R}^m$, and $\underbrace{\text{rank}(A)=m}_{\text{"full row-rank"}}$, $m\leq n$, is a polyhedron in **standard form**.

How do we find basic solutions of polyhedra in standard form?

$A\bar{x}=\bar{b}$ gives m LI constraints, as $\text{rank}(A)=m$. Hence we need $n-m$ more binding constraints from $\bar{x}\geq\bar{0}$ (i.e., $x_j\geq 0$). Which $x_j\geq 0$ should we choose?

We cannot pick just any $(n-m)$ x_j's! The following theorem describes how to pick them.

Theorem 2.4 (BT-ILO)

$\bar{x}^*\in\mathbb{R}^n$ is a **basic solution** of P in standard form iff $A\bar{x}^*=\bar{b}$, and there exist indices $\underbrace{B(1),\ldots,B(m)}_{\text{numbers in } \{1,\ldots,n\}}$ such that

(a) the columns $A_{B(1)},\ldots,A_{B(m)}$ are LI, and

(b) if $i\neq B(1),\ldots,B(m)$, then $x_i^*=0$.

Based on this theorem, we could describe a procedure for constructing basic solutions of a polyhedron in standard form.

Procedure for constructing basic solutions

1. Choose m LI columns $A_{B(1)}, \dots, A_{B(m)}$.
2. Set $x_i = 0$ for $i \neq B(1), \dots, B(m)$,
3. Solve $A\bar{x} = \bar{b}$ for unknown $x_{B(1)}, \dots, x_{B(m)}$.
   If $\bar{x} \geq \bar{0}$, then it is a bfs.

We make the following observations.

* $x_{B(1)}, \dots, x_{B(m)}$ are basic variables.
* $A_{B(1)}, \dots, A_{B(m)}$ (the corresponding columns of A) are **basic columns**, they are LI and span $\mathbb{R}^m$.
* let $B = \begin{bmatrix} A_{B(1)} & A_{B(2)} & \cdots & A_{B(m)} \end{bmatrix}$, $\bar{x}_B = \begin{bmatrix} x_{B(1)} \\ \vdots \\ x_{B(m)} \end{bmatrix}$, $\bar{x}_N \to$ remaining x_i's

  $\hookrightarrow$ basis matrix, is invertible. we set $\bar{x}_N = \bar{0}$.

* We can "split" the original system $A\bar{x} = \bar{b}$ as follows:
  We write $A = [B\ N]$, where N are the (remaining) nonbasic columns,

  and $\bar{x} = \begin{bmatrix} \bar{x}_B \\ \bar{x}_N \end{bmatrix}$.

We will finish the discussion in the next lecture...

Procedure for constructing basic solutions (continued)

* With $A = [B\ N]$, $\bar{x} = \begin{bmatrix} \bar{x}_B \\ \bar{x}_N \end{bmatrix}$, $A\bar{x} = \bar{b}$ is equivalent to

$$[B\ N]\begin{bmatrix} \bar{x}_B \\ \bar{x}_N \end{bmatrix} = \bar{b}$$

$$\Rightarrow \quad B\bar{x}_B + N\bar{x}_N = \bar{b}$$

We then set $\bar{x}_N = \bar{0}$.

$$\Rightarrow \quad B\bar{x}_B = \bar{b} \quad \Rightarrow \quad \bar{x}_B = B^{-1}\bar{b}$$ (recall, B is invertible)

$$\left.\begin{aligned} x_1 + x_2 - x_3 \qquad\quad &= 2 \\ 3x_1 + x_2 \qquad - x_4 \quad &= 4 \\ 3x_1 + 2x_2 \qquad\quad + x_5 &= 10 \end{aligned}\right\} \quad A = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 3 & 1 & 0 & -1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 2 \\ 4 \\ 10 \end{bmatrix}$$

$$x_j \geq 0 \ \forall j$$

Correspondence between corner pts & bfs's

We can use the above procedure to identify the bfs's in the standard form that correspond to the corner points marked out in the 2D graphical representation (of the original polyhedron). The maximum number of basic solutions possible in this case is $\binom{n}{m} = \binom{5}{3} = 10$. In this example, it indeed is true that each choice of $m=3$ columns of A produce a basic solution. In a general case, not all of the $\binom{n}{m}$ choices of m columns may produce a basis (i.e., an invertible matrix B). Hence, the number of basic solutions may be smaller than $\binom{n}{m}$.

Also, note that not all of these $\binom{n}{m}$ bases may lead to bfs's.
We can use MATLAB to demonstrate the correspondences between the bfs's (and basic solutions) in the standard form LP and the corner points in the 2D picture. See the course web page for the MATLAB files.

e.g., $B(1)=1, B(2)=2, B(3)=3$ gives $B = A(\underset{\text{all rows}}{:}, \underset{\text{columns } 1,2,3}{[1\ 2\ 3]})$

$$B = \begin{bmatrix} 1 & 1 & -1 \\ 3 & 1 & 0 \\ 3 & 2 & 0 \end{bmatrix}, \quad \det(B) = -3 \quad \text{(hence } B \text{ is invertible).}$$

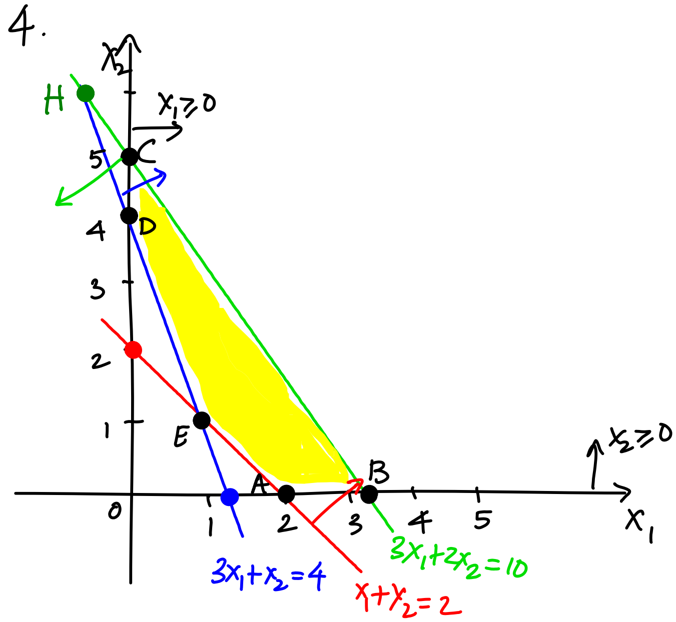
We set $x_4 = x_5 = 0$ $(\bar{x}_N = \bar{0})$, and solve for $\bar{x}_B = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = B^{-1}\bar{b}$, to get

$\bar{x}_B = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2/3 \\ 6 \\ 10/3 \end{bmatrix}$. This point is not feasible (as all entries are not ≥ 0); hence it is not a bfs.

This vector corresponds to the vertex $H(-2/3, 6)$ in the picture.

As another example, consider $B(1)=2, B(2)=3, B(3)=4$.

$B = A(:,[2\ 3\ 4]) = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 2 & 0 & 0 \end{bmatrix}$. $\det(B) = 2$, hence B is invertible. We get



$$\bar{x}_B = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} = B^{-1}\bar{b} = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix}. \text{ Thus,}$$

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 3 \\ 1 \\ 0 \end{bmatrix} \text{ is a bfs.}$$

It corresponds to the vertex $C(0,5)$.

MATH 464 – Lecture 10 (02/09/2023)

Today: * correspondence between bfs's and vertices
* degenerate bfs.

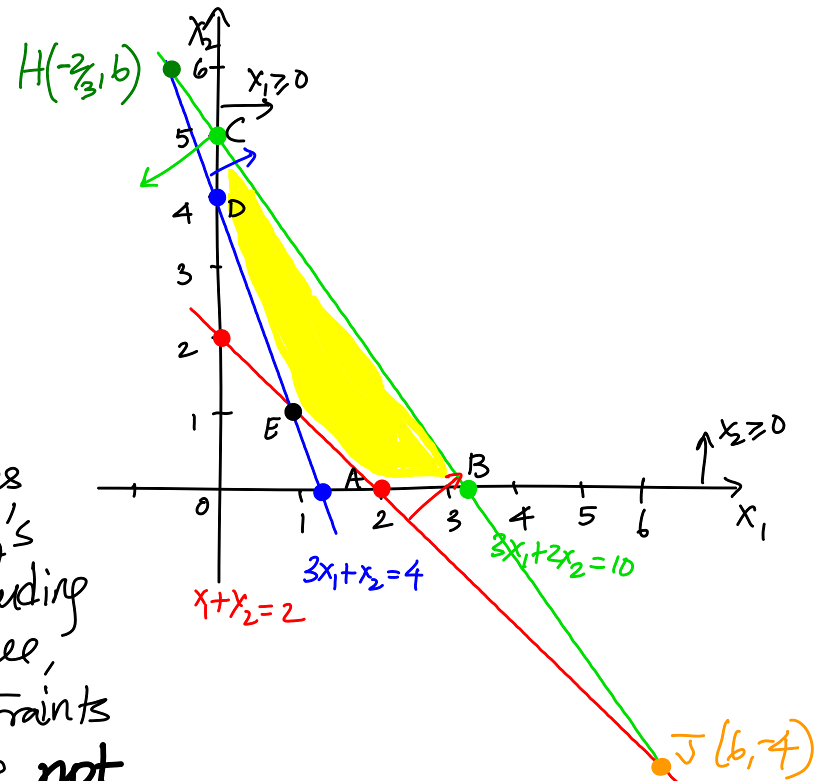
Recall: Procedure to construct basic solutions and bfs's

With $A = [B \; N]$, $\bar{x} = \begin{bmatrix} \bar{x}_B \\ \bar{x}_N \end{bmatrix}$, $A\bar{x} = \bar{b} \implies B\bar{x}_B + N\bar{x}_N = \bar{b}$

We set $\bar{x}_N = \bar{0} \implies B\bar{x}_B = \bar{b} \implies \bar{x}_B = B^{-1}\bar{b}$.

$$\left.\begin{array}{l} x_1 + x_2 - x_3 \qquad\quad = 2 \\ 3x_1 + x_2 \quad - x_4 \quad = 4 \\ 3x_1 + 2x_2 \qquad + x_5 = 10 \end{array}\right\}$$
$x_j \geq 0 \; \forall j$

$$A = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 3 & 1 & 0 & -1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{bmatrix}_{3\times 5} \quad \bar{b} = \begin{bmatrix} 2 \\ 4 \\ 10 \end{bmatrix}$$



We could identify the basis indices at any vertex by identifying the x_j's that are >0 at that vertex, including slack/excess variables. For instance, at $A(2,0)$, $x_1 > 0$. Also, the constraints $3x_1 + x_2 \geq 4$ and $3x_1 + 2x_2 \leq 10$ are **not** active at A, and hence $x_4 > 0$ and $x_5 > 0$. So $B_{ind} = [B(1) \; B(2) \; B(3)] = [1 \; 4 \; 5]$ at the bfs $\equiv$ A.

We can use the same method to identify basic solutions for $H(-\frac{2}{3}, 6)$ and $J(6,-4)$. At H, x_1, x_2, and x_3 are $\neq 0$ (not all of them are >0 here!). Hence $B_{ind} = [1 \; 2 \; 3]$ at H.

as they're not feasible!

Similarly, $B_{ind} = [1 \; 2 \; 4]$ at J.

See MATLAB session for details.

We can find the bfs in the standard form LP corresponding to each corner point by inspection. In other words, we do not necessarily have to enumerate all $\binom{n}{m}$ choices for bases.

For instance, at the vertex $E(1,1)$, the constraints $x_1+x_2 \geq 2$ and $3x_1+x_2 \geq 4$ are active, while $3x_1+2x_2 \leq 10$ is not binding. Hence $x_3=x_4=0$ and $x_5>0$ in the corresponding bfs from the standard form LP. Also notice that both x_1 and x_2 are >0. Hence, $\{x_1, x_2, x_5\}$ is indeed the corresponding basis.

Similarly, at $C(0,5)$, we see that both $x_1+x_2 \geq 2$ and $3x_1+x_2 \geq 4$ are not active, while $3x_1+2x_2 \leq 10$ and $x_1 \geq 0$ are binding. Hence $x_3>0$ and $x_4>0$, but $x_5=0$. Also, $x_2=5$ is >0. Thus $\{2,3,4\}$ is the set of basic columns.

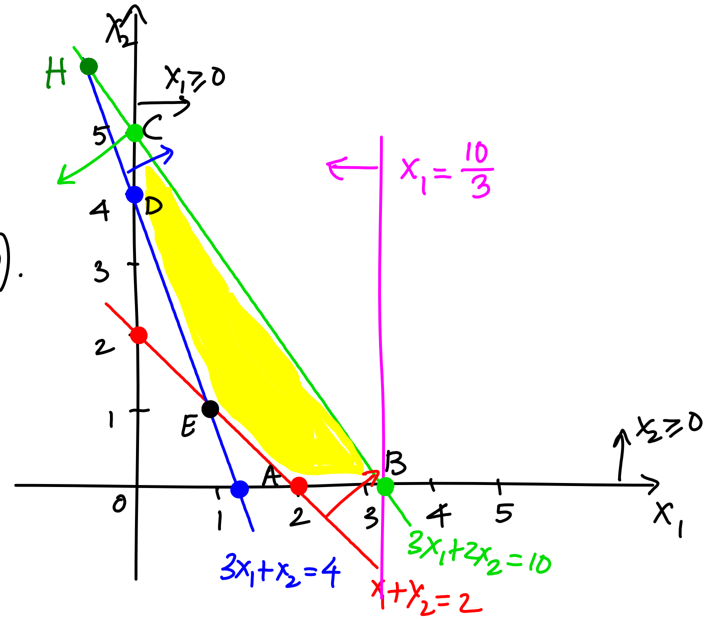
The following table summarizes the correspondences between the bfs's in the standard form polyhedron and the vertices identified in the 2D picture.

$\det(B) \neq 0$ confirms that the basis matrix is invertible in each instance.

Vertex	B(1)	B(2)	B(3)	det(B)	x_1	x_2	x_3	x_4	x_5
					BFS				
A	1	4	5	-1	2	0	0	2	4
B	1	3	4	3	10/3	0	4/3	6	0
C	2	3	4	2	0	5	3	1	0
D	2	3	5	1	0	4	2	0	2
E	1	2	5	-2	1	1	0	0	5

Degeneracy

Recall adding $x_1 \le \frac{10}{3}$ to the original LP:



Now, the three constraints $3x_1+2x_2 \le 10$, $x_1 \le \frac{10}{3}$ and $x_2 \geqslant 0$ are active at $B(\frac{10}{3}, 0)$.

We need only two LI constraints to define a BFS here. Hence, B is a degenerate bfs.

In the standard form, we would have $x_1 + x_6 = \frac{10}{3}$, where x_6 is the slack variable for the constraint $x_1 \le \frac{10}{3}$. Notice that we get $x_2 = x_5 = x_6 = 0$ now. We have $n=6$ and $m=4$, and hence more than $n-m=2$ x_j's are zero. This is an illustration of the condition for a bfs to be degenerate in general.

Def (Degeneracy in Standard form) A basic solution $\bar{x}$ of P in standard form is **degenerate** if more than $n-m$ x_j's are zero.

But we could have identified B as a degenerate vertex from the 2D picture itself, from the fact that more than $n (=2)$ LI constraints (3 here) are active at B. Indeed, this condition holds more generally as well.

Def A basic solution $\bar{x}^* \in \mathbb{R}^n$ of P (in general form) is a **degenerate basic solution** if more than n constraints are active at $\bar{x}^*$.

A degenerate basic solution that is feasible is called a **degenerate bfs**.

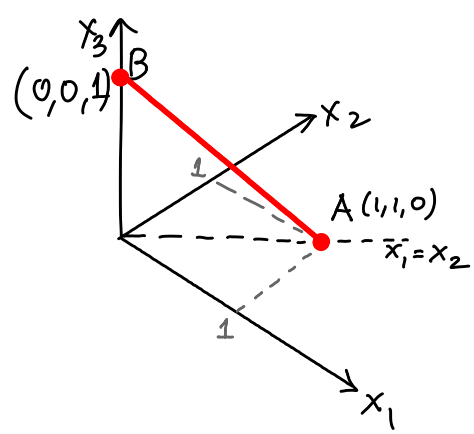
* In 2D, a basic solution has 3 or more lines corresponding to constraints satisfied as equations meeting at the point.
(including $x_j \geq 0$)

* In $\mathbb{R}^n$, we need $n+1$ or more constraints active at a degenerate basic solution. Hence, if $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \geq \bar{b}\}$, $A_{m \times n}$, $m \leq n$, there are no degenerate basic solutions.

Degeneracy may not be much of a problem in small instances. But when we are solving large instances of LP, it could create inefficiencies. The typical algorithm tries to move from one bfs to an adjacent bfs such that the objective function value improves, or at least does not become worse. It could happen that we cycle through several degenerate bfs's before ultimately jumping to a vertex that actually strictly improves the objective function value.

But as we are going to see, degeneracy depends very much on how we represent the polyhedron. Indeed, we could throw out $x_1 \leq 1/3$ without changing the polyhedron, and remove the degeneracy at B. Here is another example.

Consider the polyhedron $P = \{\bar{x} \in \mathbb{R}^3 \mid x_1 - x_2 = 0,\ x_1 + x_2 + 2x_3 = 2;\ x_j \geq 0, j = 1,2,3\}$ with $m = 2$, $n = 3$.



A (1,1,0) is a non-degenerate vertex, but $B(0,0,1)$ is a degenerate bfs.

$x_1-x_2=0$, $x_1+x_2+2x_3=2$, and $x_3\geq 0$ are active at $A(1,1,0)$.
Indeed, these three constraints are LI. The corresponding vectors of the constraints are $\begin{bmatrix}1\\-1\\0\end{bmatrix}$, $\begin{bmatrix}1\\1\\2\end{bmatrix}$, and $\begin{bmatrix}0\\0\\1\end{bmatrix}$.

$x_1-x_2=0 \equiv [1\ \ -1\ \ 0]\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=0$; $\quad x_1+x_2+2x_3=2 \equiv [1\ \ 1\ \ 2]\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=2$.

At $B(0,0,1)$, $x_1-x_2=0$, $x_1+x_2+2x_3=2$, $x_1\geq 0$, and $x_2\geq 0$ are active.
Of course, the four constraints in 3D are not LI.

Alternatively, we can describe the same polyhedron as

$$P=\{\bar{x}\in\mathbb{R}^3 \mid x_1-x_2=0,\ x_1+x_2+2x_3=2;\ x_1\geq 0,\ x_3\geq 0\}.$$ no $x_2 \geq 0$.

For this description, $B(0,0,1)$ is NOT degenerate, as now we have only 3 active constraints at B, namely, $x_1-x_2=0$, $x_1+x_2+2x_3=2$, and $x_1\geq 0$.

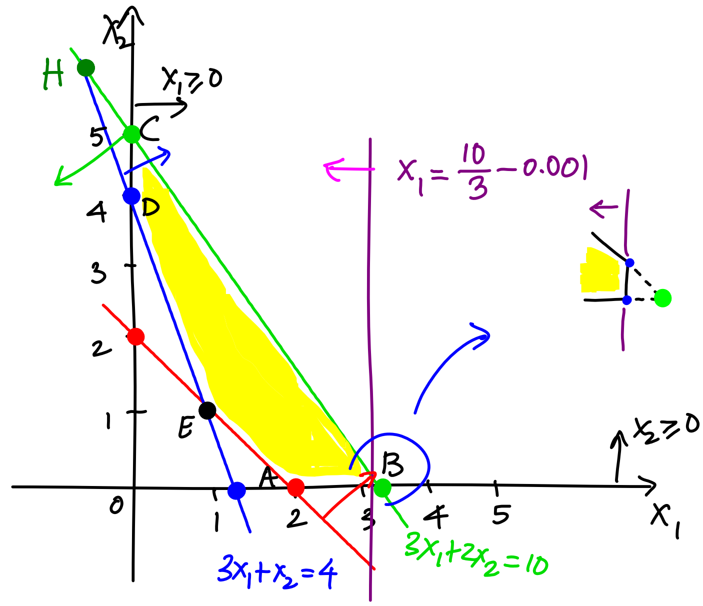
But if we replace $x_1+x_2+2x_3=1$ now with two inequalities $x_1+x_2+2x_3\geq 1$ and $x_1+x_2+2x_3\leq 1$, B is degenerate again!
So, $B(0,0,1)$ is a degenerate bfs of

$$P=\{\bar{x}\in\mathbb{R}^3 \mid x_1-x_2=0,\ x_1+x_2+2x_3\geq 1,\ x_1+x_2+2x_3\leq 1,\ x_1\geq 0,\ x_3\geq 0\}.$$

constraints active at B

If permitted, we could avoid degeneracy by perturbing some of the constraint a tiny bit.

So, $x_1 \geq \frac{10}{3}$ could be replaced by $x_1 \geq \frac{10}{3} - 0.001$.



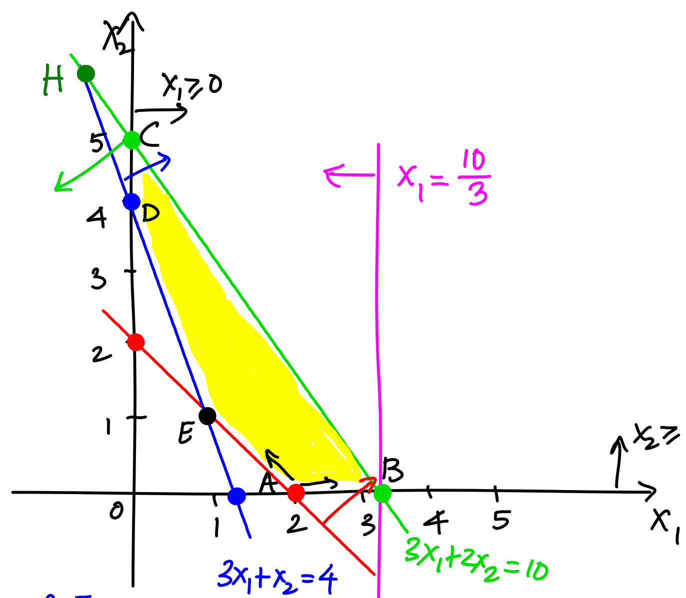
But whether we could do such minor modifications will depend very much on the specific application in question!

Let's go back to our example, and identify degenerate bfs's:

Recall that a basic solution $\bar{x} \in P$ in standard form is degenerate if more than $n-m$ x_j's are zero.

$$
\begin{aligned}
x_1 + x_2 - x_3 \quad\quad\quad &= 2\\
3x_1 + x_2 \quad - x_4 \quad\quad &= 4\\
3x_1 + 2x_2 \quad\quad + x_5 \quad &= 10\\
x_1 \quad\quad\quad\quad + x_6 &= 10/3
\end{aligned}
$$

$x_j \geq 0 \;\; \forall j$



original A

original $\bar{b}$

$$
A = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 \\ 3 & 1 & 0 & -1 & 0 & 0 \\ 3 & 2 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{4\times 6} \qquad \bar{b} = \begin{bmatrix} 2 \\ 4 \\ 10 \\ 10/3 \end{bmatrix}
$$

$\text{rank}(A) = m = 4.$

Indeed, rank$(A) = 4 = m$ now. So we are still in the original setting of standard form polyhedron. Let's figure out the bfs corresponding to the vertex B. Notice that $\{x_1, x_3, x_4\}$ are all basic, i.e., > 0 at B. We need one more x_j out of $x_2, x_5, x_6)$ to complete a basis. Which one could we select?

With $B(1)=1, B(2)=2, B(3)=3, B(4)=4$, we get $B = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 3 & 1 & 0 & -1 \\ 3 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$, $\det(B) = -2$,

So B is indeed invertible. We get $\bar{x}_B = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = B^{-1}\bar{b} = \begin{bmatrix} 10/3 \\ 0 \\ 4/3 \\ 6 \end{bmatrix}$.

In fact, we could pick any one of $\{x_2, x_5, x_6\}$ with $\{x_1, x_3, x_4\}$ to get a basis here. Check the Matlab session for details. In each case, we get 3 x_j's set at 0. Recall that $n-m = 6-4 = 2$ here, thus showing that more than $n-m$ x_j's are zero at the degenerate bfs.

including the non-basic x_j's

Qn. If there are $k > n$ active constraints, is it always true that there are $\binom{k}{k-n} = \binom{k}{n}$ different bases that lead to the same degenerate bfs? (will be in next homework)

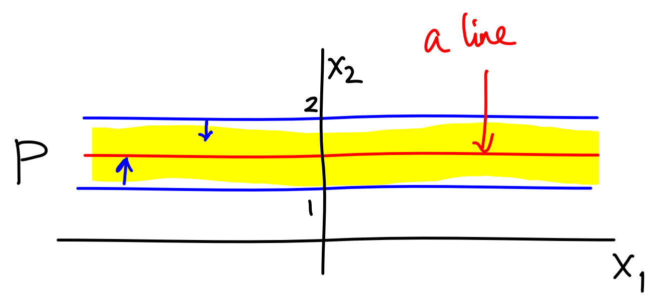
MATH464 – Lecture 11 (02/14/2023)

Today: * properties of polyhedra
* feasible and basic directions

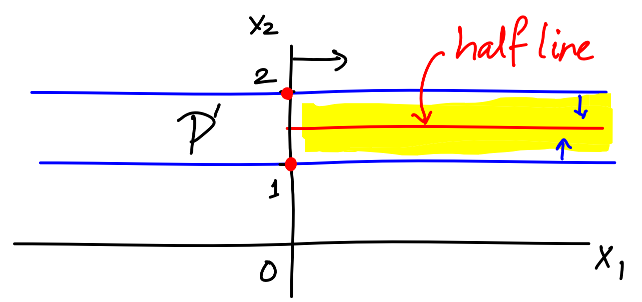
We have seen that vertices were optimal in many of the LP instances we solved. Indeed, we are building up the machinery to talk about the simplex algorithm, which will move from one bfs to an adjacent bfs in each step. But we first study some properties of the polyhedron in general.

Does every polyhedron have an extreme point?

$P = \{\bar{x} \in \mathbb{R}^2 \mid 1 \leq x_2 \leq 2\}$ has no corner points.



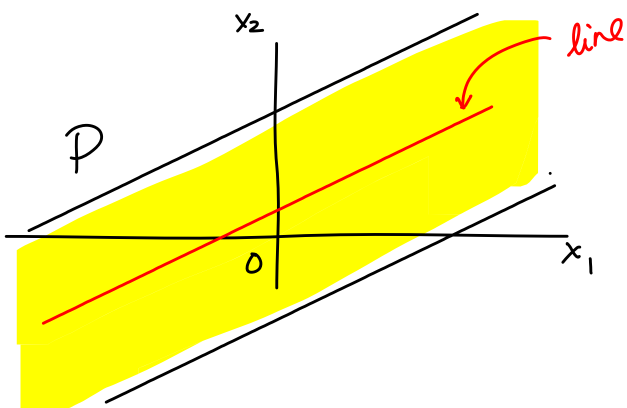
But if we were to add $x_1 \geqslant 0$ to make in P', we get two corner points.

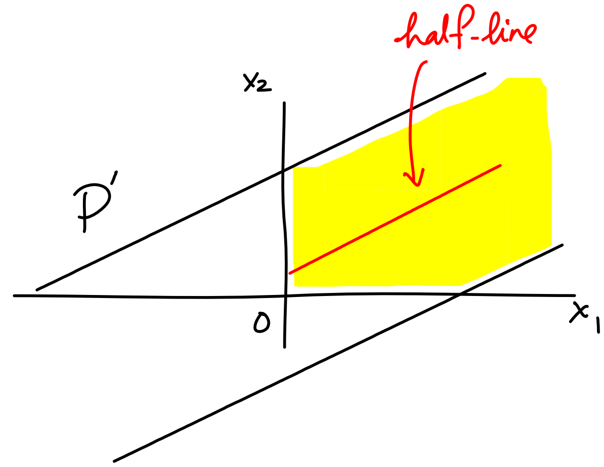


$P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \geqslant \bar{b}\}$ with $A_{m \times n}$ and $m < n$ cannot have any basic solutions, and hence cannot have any bfs's!

We make the following observation. P has a line, which is parallel to the two constraint lines (which themselves are parallel to each other). But P' (with $x_1 \geqslant 0$) can have only a half line (and not a line).

Intuitively, P is "corner-free" if it contains a line, which extends without limit in either direction. Another example: P on the left a line, and hence has no vertices.

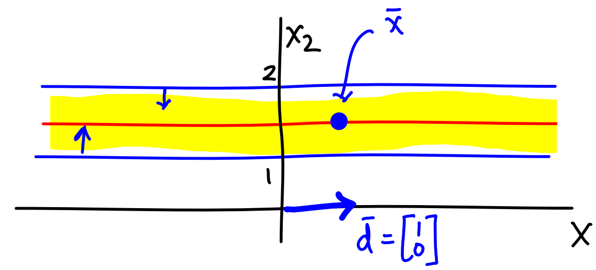




But P does not contain a line once $x_j \geq 0$ is added (to get P').

We formalize the notion of a set containing a line, and the intuition that a polyhedron without a line has a vertex.

Def $P \subset \mathbb{R}^n$ contains a line if there exists some $\bar{x} \in P$ and a direction $\bar{d} \in \mathbb{R}^n$, $\bar{d} \neq \bar{0}$, such that $\bar{x} + \lambda \bar{d} \in P \;\forall\; \lambda \in \mathbb{R}$.



Theorem 2.6 BT-ILO Let $P = \{\bar{x} \in \mathbb{R}^n \mid \bar{a}_i^T \bar{x} \geq b_i,\ i = 1, \ldots, m\}$, $P \neq \emptyset$. $m \geq n$

The following statements are equivalent.

(i) P has at least one extreme point.

(ii) P does not contain a line.

(iii) There are n vectors in $\{\bar{a}_1, \ldots, \bar{a}_m\}$ which are LI.

We had already noted that if $m < n$, P has no bfs.

We immediately get the following corollary.

BT-ILO Corollary 2.2 Every nonempty bounded polyhedron, and every polyhedron in standard form, has at least one bfs.

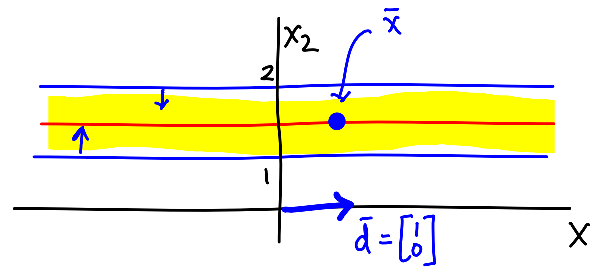
If P is bounded, it cannot contain a line, so (ii) holds.

Notice that $\bar{a}_i^T\bar{x} \geq b_i$, $i=1,\ldots,m$ above include $x_j \geq 0 \; \forall j$. Indeed, as soon as we have added $x_j \geq 0 \; \forall j$, we do have n $\bar{a}_i$'s that are LI — the unit vectors corresponding to $x_j \geq 0$, i.e., $\bar{a}_i = \bar{e}_i$, the i^{th} unit vector. Hence (iii) holds.

Qn. If P has no corner points, does the LP

$$\min \{\bar{c}^T\bar{x} \mid \bar{x} \in P\}$$

have any optimal solutions?



e.g., $\min \{x_2 \mid 1 \leq x_2 \leq 2\}$ has alternative optimal solutions (any point on the line $x_2=1$ is optimal).

But $\min \{x_1+x_2 \mid 1 \leq x_2 \leq 2\}$ is an unbounded LP.

At the same time, we could make the following statement.

If P has no corner points, then the LP **cannot** have a unique optimal solution. The next theorem formalizes the reverse implication.

BT-ILO Theorem 2.7 Consider $\min\{\bar{c}^T\bar{x} \mid \bar{x} \in P\}$.
If P has at least one extreme point, and if the LP has an optimal solution, then there exists an extreme point which is an optimal solution.

We could get a slightly more general result, which specifies what happens when there is no optimal solution.

BT-ILO Theorem 2.8 Consider $\min\{\bar{c}^T\bar{x} \mid \bar{x} \in P\}$, where P is a polyhedron with at least one extreme point. Then the optimal cost is $-\infty$ or there exists an extreme point which is optimal.

See BT-ILO for the proofs.

Notice that the above theorems cover the case of polyhedra with at least one extreme point. But what about polyhedra without extreme points? Indeed, we can generalize the above theorem to get the following corollary. In particular, recall $\min\{x_2 \mid 1 \le x_2 \le 2\}$.

(Corollary 2.3 BT-ILO) Consider $\min\{\bar{c}^T\bar{x} \mid \bar{x} \in P\}$, where P is a nonempty polyhedron. Then the optimal cost is $-\infty$, or the exists an optimal solution.

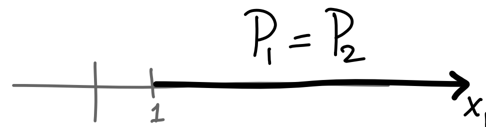
This result might not hold for non-linear problems.

consider $\left\{ \begin{array}{l} \min \frac{1}{x} \\ \text{s.t. } x \geq 1 \end{array} \right\}$ There is no optimal solution here, but the optimal cost is not $-\infty$.

We are now ready to present the simplex method to solve LPs in general dimensions. We will present this algorithm for LPs in standard form. In this context, we consider one more aspect of the standard form.

Qn. Does the "shape" of a polyhedron change when we convert it to standard form?

Consider $P_1 = \{x_1 \in \mathbb{R} \mid x_1 \geq 1\}$.



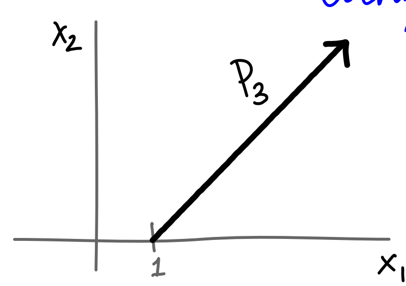
We add nonnegativity to get

$P_2 = \{x_1 \in \mathbb{R} \mid x_1 \geq 1, x_1 \geq 0\}$. Notice that $P_2 = P_1$.

Indeed, $x_1 \geq 0$ is redundant since $x_1 \geq 1$ is given. But we're just following the procedure here (to eventually convert P_1 to standard form).

We convert P_2 to standard form to get

$P_3 = \{\bar{x} \in \mathbb{R}^2 \mid x_1 - x_2 = 1, x_1, x_2 \geq 0\}$.

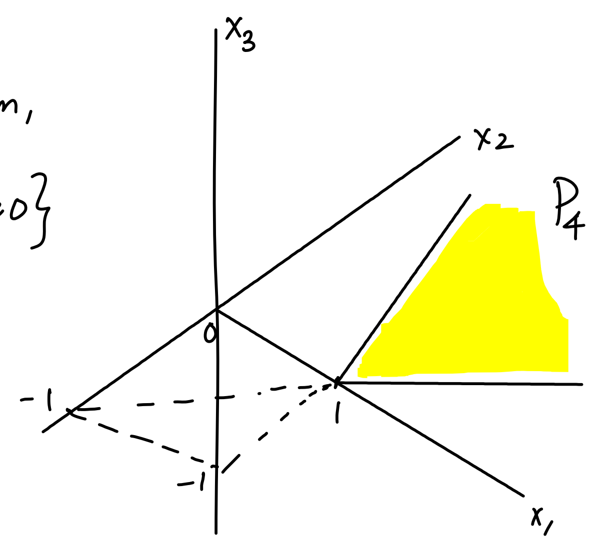


P_3 looks very much like P_1 (or P_2). They are all half-lines (or rays).

Now, if we convert P_1 to standard form, we get

$$P_4 = \{\bar{x} \in \mathbb{R}^3 \mid \underbrace{(x_1 - x_2)}_{= x_1 \text{ in } P_1} - x_3 = 1, x_j \geq 0\}$$

x_1 urs



P_4 is a portion of a plane in the nonnegative orthant in $\mathbb{R}^3$. Notice the similarity to P_1, P_2, and P_3.

Simplex Method (Chapter 3 in BT-ILO)

The simplex method generalizes the graphical solution method in 2D to higher dimensions. It moves from one bfs (vertex) to an adjacent bfs (vertex) where the objective function improves.

We will describe the method for an LP in standard form where $A \in \mathbb{R}^{m\times n}$, $\bar{b} \in \mathbb{R}^m$, $\text{rank}(A) = m$, $m \leq n$.

$$\left\{ \begin{array}{ll} \min & \bar{c}^T\bar{x} \\ \text{s.t.} & A\bar{x} = \bar{b} \\ & \bar{x} \geqslant \bar{0} \end{array} \right\}.$$

We want to define **optimality conditions** for $\bar{x} \in P$. If these conditions are satisfied, then $\bar{x}$ is an optimal solution.

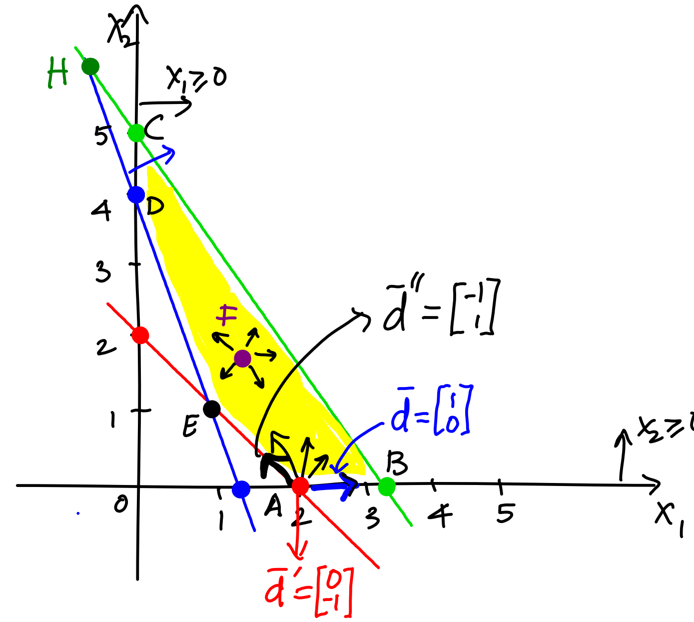
Algorithm: Given some $\bar{x} \in P$, check optimality conditions. If they are not satisfied, we search "nearby" to see if we can improve, i.e., decrease, $\bar{c}^T\bar{x}$.

In 1D calculus, the optimality conditions for $x \in \mathbb{R}$ to be a local minimum of $f(x)$ are $f'(x) = 0$ and $f''(x) > 0$. If $f(x)$ is a convex function, these conditions also guarantee that x is a global minimum.

For LPs, since $f(\bar{x}) = \bar{c}^T\bar{x}$ is linear, and since P is a polyhedron (hence convex), a local optimum is also a global optimum.

When searching "nearby", we want to make sure we always stay feasible, i.e., we do not want to go outside the feasible region.

Suppose we are at $A(2,0)$. We can consider directions to move. If we move straight down, i.e., along $\bar{d}' = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$, we will go outside the feasible region. But $d = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ is a good direction — we can move right all the way up to $B(10/3, 0)$.



In general, from $\bar{x}$, we consider $\bar{x} + \theta \bar{d}$ for $\theta > 0$ to move along $\bar{d}$. We want $\bar{x} + \theta \bar{d} \in P$.

Thus, no $\theta > 0$ exists for $\bar{d}'$, while any $0 < \theta \leq \frac{4}{3}$ works for moving along $\bar{d}$. Similarly, if we move Northwest, i.e., along $\bar{d}' = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, we can go all the way to $E(1,1)$. So any $0 < \theta \leq 1$ works. Further, all directions "in between" $d = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $d'' = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ are good — see figure.

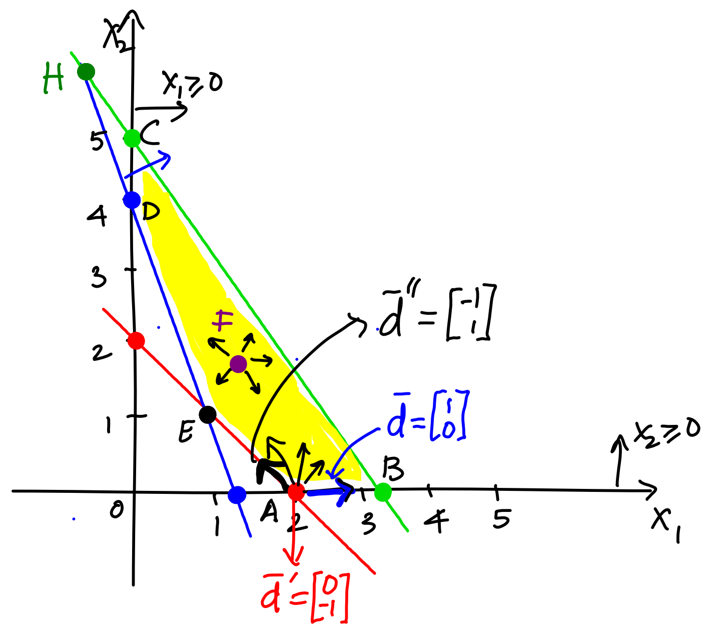
If F is in the interior of P, then any $\bar{d}$ is a good direction (see figure).

We formalize this notion of a "good" direction. (in the next lecture...)

MATH 464 – Lecture 12 (02/16/2023)

Today:
* basic direction
* reduced cost
* optimality conditions

Recall the notion of "good directions" to move from any point $\bar{x} \in P$. We want to stay feasible – and improve the objective function in the process.



We formalize this notion of a "good" direction now.

Def Let $\bar{x} \in P$. A vector $\bar{d} \in \mathbb{R}^n$ is a **feasible direction** at $\bar{x}$ if there exists $\theta > 0$ such that $\bar{x} + \theta \bar{d} \in P$.

Notice that θ can be arbitrarily small, as long as it is > 0.

For instance, at $\bar{x} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, $\bar{d}' = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ is not a feasible direction, as

$$\bar{x} + \theta \bar{d}' = \begin{bmatrix} 2 + 0\theta \\ 0 + (-1)\theta \end{bmatrix} \overset{A}{=} \begin{bmatrix} 2 \\ -\theta \end{bmatrix} \notin P \text{ for any } \theta > 0.$$

As we can see, there could be many feasible directions at a bfs. How do we choose a good one? And how far do we move along that direction?

We now describe how to move from one bfs to an adjacent bfs using (linear) algebra — how do we actually implement the "move"?

Let $\bar{x}$ be a bfs for $\underbrace{B(1), \ldots, B(m)}_{\text{basis}}$. With $\bar{x}_B = \begin{bmatrix} x_{B(1)} \\ \vdots \\ x_{B(m)} \end{bmatrix}$, and $\bar{x}_N$ collecting the remaining nonbasic variables, we set $B = [A_{B(1)} \cdots A_{B(m)}]$ as the basis matrix (made of the basic columns of A). We get that B^{-1} exists, and after setting $\bar{x}_N = \bar{0}$, we can find $\bar{x}_B = B^{-1}\bar{b}$.

Consider the direction $\bar{d}$ at $\bar{x}$ such that $d_j = 1$ for a non-basic x_j, and $d_i = 0$ for $i \neq B(1), \ldots, B(m), j$. all other non basic d_i's are set to zero.

Let $\bar{d}_B = \underbrace{\begin{bmatrix} d_{B(1)} \\ \vdots \\ d_{B(m)} \end{bmatrix}}_{\text{to be found}}$. $\bar{d}_N = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ corresponding to x_j

We want to move to $\bar{x} + \theta \bar{d}$, $\theta > 0$, and stay feasible. So we need $A(\bar{x} + \theta \bar{d}) = \bar{b}$, and $\bar{x} + \theta \bar{d} \geq 0$.

But $A\bar{x} = \bar{b}$, so we get $\theta(A\bar{d}) = \bar{0}$, i.e., $A\bar{d} = \bar{0}$, which can be written as

$$B\bar{d}_B + \sum_{i \notin B(1), \ldots, B(m)}' A_i d_i = \bar{0}, \text{ i.e., } B\bar{d}_B + A_j = \bar{0}.$$

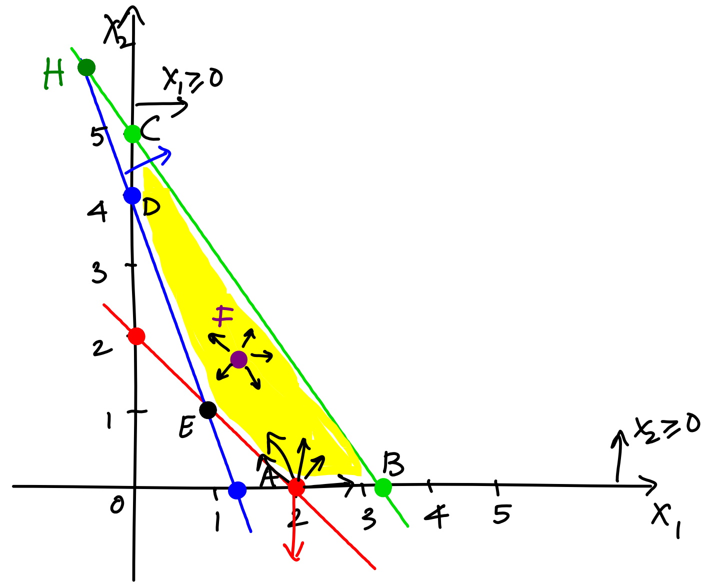
A_j: j^{th} column of A

so $$\boxed{\bar{d}_B = -B^{-1}A_j}$$

This $\bar{d}_B$ defines the **j^{th} basic direction at bfs $\bar{x}$**.

Notice that there are $(n-m)$ basic directions at a bfs $\bar{x}$.

Back to our example:



$$A = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 3 & 1 & 0 & -1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 2 \\ 4 \\ 10 \end{bmatrix}$$

BFS corresponding to $A(2,0)$.

$\{B(1), B(2), B(3)\} = \{1, 4, 5\}$

$$B = A(:, [1\,4\,5]) = \begin{bmatrix} 1 & 0 & 0 \\ 3 & -1 & 0 \\ 3 & 0 & 1 \end{bmatrix}$$

$j=2$: $\quad A_2 = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \quad \bar{d}_B = -B^{-1}A_2 = \begin{array}{c} \begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix} \end{array}\begin{array}{c} 1 \\ 4 \\ 5 \end{array} \Rightarrow \bar{d} = \begin{bmatrix} -1 \\ 1 \\ 0 \\ -2 \\ 1 \end{bmatrix}$ move toward E along $\overline{AE}$

$d_2 = 1, \ d_3 = 0$

$j=3$: $\quad d_3 = 1, \ d_2 = 0 \quad \bar{d}_B = -B^{-1}A_3 = \begin{array}{c} \begin{bmatrix} 1 \\ 3 \\ -3 \end{bmatrix} \end{array}\begin{array}{c} 1 \\ 4 \\ 5 \end{array} \Rightarrow \bar{d} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 3 \\ -3 \end{bmatrix}$ move toward B along $\overline{AB}$

See Matlab session on the course web page for details.

When we move along $\bar{d}$ from $\bar{x}$ to $\bar{x} + \theta\bar{d}$, $A\bar{x} = \bar{b}$ is satisfied.

How about $\bar{x} \geq \bar{0}$? For the nonbasic variables, $x_j > 0$,

and $x_i = 0$ for $i \neq j$, i nonbasic. What about $\bar{x}_B$?

(a) If $\bar{x}$ is non-degenerate, i.e., $\bar{x}_B > 0$, $\bar{x}_B + \theta \bar{d}_B \geq \bar{0}$ as long as θ is sufficiently small.

(b) If $\bar{x}$ is degenerate, a good $\theta > 0$ may not exist, as there might be some $B(i)$ such that $x_{B(i)} = 0$ and $d_{B(i)} = -1$, which will make $x_{B(i)} = -\theta < 0$!

We will describe the non-degenerate case first.

Now, let's incorporate the objective function $\min \bar{c}^T \bar{x}$. Let $\bar{d}$ be the j^{th} basic direction. We move from $\bar{x}$ to $\bar{x} + \theta \bar{d}$, and observe how $\bar{c}^T \bar{x}$ changes, i.e., from $\bar{c}^T \bar{x}$ to $\bar{c}^T(\bar{x} + \theta \bar{d}) = \bar{c}^T \bar{x} + \theta (\bar{c}^T \bar{d})$.

$\bar{c}^T \bar{d}$ is the rate of change of the cost function when moving along $\bar{d}$.

$$\bar{c}^T \bar{d} = \bar{c}_B^T \bar{d}_B + \bar{c}_N^T \bar{d}_N = \bar{c}_B^T \bar{d}_B + c_j \quad \text{as } \bar{d}_N = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow j^{th} \text{ entry}$$

$$= \bar{c}_B^T(-B^{-1}A_j) + c_j$$

$$= c_j - \bar{c}_B^T B^{-1} A_j$$

$$c_j' = \underbrace{c_j}_{\text{cost per unit increase of } x_j} - \underbrace{\bar{c}_B^T B^{-1} A_j}_{\text{compensating change so that } A\bar{x}=\bar{b} \text{ is still satisfied — reduce the cost by this amount.}}$$ is the **reduced cost** of x_j.

book uses $\bar{c}$

In vector form $\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A$.

In particular, for the basic columns, we get

$$\bar{c}_B'^T = \bar{c}_B^T - \bar{c}_B^T \underbrace{B^{-1} B}_{I} = \bar{c}_B^T - \bar{c}_B^T = \bar{0}.$$

Thus, the reduced cost of every basic variable is zero.

e.g., At $A(2,0)$, the bfs in standard form is $\bar{x} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 2 \\ 4 \end{bmatrix}$,

and the reduced cost is $\bar{c}' = \begin{bmatrix} 0 \\ -1 \\ 2 \\ 0 \\ 0 \end{bmatrix}$.

Since $c_2' = -1$, if we move along the 2^{nd} basic direction, the cost will decrease.

If $\bar{c}' \geq \bar{0}$, then none of the basic directions could give an improvement in $\bar{c}^T \bar{x}$. In other words, the solution is optimal!

We describe this condition ($\bar{c}' \geq \bar{0}$) as the optimality condition for LP (in the next lecture).

MATH 464 – Lecture 13 (02/21/2023)

Today: * LP optimality conditions
* Details of the simplex method

Recall: Reduced costs: $\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A$ (for LP in standard form: $\min\{\bar{c}^T\bar{x} \mid A\bar{x}=\bar{b}, \bar{x}\geqslant\bar{0}\}$ with $A \in \mathbb{R}^{m\times n}$, rank$(A)=m$, $m\leq n$.

BT-ILO Theorem 3.1 $\bar{x}$ is a bfs, B: basis matrix, $\bar{c}'$ reduced costs.

(a) If $\bar{c}'\geqslant\bar{0}$, then $\bar{x}$ is optimal

(b) If $\bar{x}$ is nondegenerate and optimal, then $\bar{c}'\geqslant\bar{0}$.

Proof (a) Let $\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A \geqslant \bar{0}^T$. We want to show $\underline{\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y}}$ $\forall \bar{y}\in P$ (i.e., $\bar{x}$ is optimal).

Let $\bar{y}\in P$ be an arbitrary feasible point. So, $A\bar{y}=\bar{b}, \bar{y}\geqslant\bar{0}$.

Let $\bar{d}=\bar{y}-\bar{x}$. Also, $A\bar{x}=\bar{b}$, $\bar{x}\geqslant\bar{0}$ (as $\bar{x}$ is a bfs).

Want to prove $\bar{c}^T\bar{d}\geqslant\bar{0}$

$$\Rightarrow A\bar{d} = A(\bar{y}-\bar{x}) = \bar{0}.$$

$$\Rightarrow B\bar{d}_B + N\bar{d}_N = \bar{0} \Rightarrow B^{-1}\Big(B\bar{d}_B + \sum_{i\in\mathcal{N}} A_i d_i = \bar{0}\Big).$$

$\mathcal{N}$ is the set of non-basic indices ($x_j=0$ $\forall j\in\mathcal{N}$).

$\mathcal{B}$ is the set of basis indices, i.e., $\mathcal{B}=\{B(1),\dots,B(m)\}$.

So, $\mathcal{B}\cup\mathcal{N}=\{1,\dots,n\}$.

$$\Rightarrow \bar{d}_B + \sum_{i\in\mathcal{N}} B^{-1}A_i d_i = \bar{0} \Rightarrow \bar{d}_B = -\sum_{i\in\mathcal{N}} B^{-1}A_i d_i.$$

So, $\bar{c}^T\bar{d} = \bar{c}_B^T \bar{d}_B + \bar{c}_N^T \bar{d}_N = \bar{c}_B^T \bar{d}_B + \sum_{i\in N} c_i d_i$

$$= \sum_{i\in N} c_i d_i - \sum_{i\in N} \bar{c}_B^T B^{-1} A_i d_i \qquad \bar{d}_B = -\sum_{i\in N} B^{-1} A_i d_i$$

$$= \sum_{i\in N} (c_i - \bar{c}_B^T B^{-1} A_i) d_i$$

$$= \sum_{i\in N} \underbrace{c_i'}_{\geq 0} d_i, \quad \text{as } c_i' = c_i - \bar{c}_B^T B^{-1} A_i, \text{ the } i^{th} \text{ reduced cost.}$$

We will be done if we can show $d_i \geq 0 \ \forall\, i \in N$, as $c_i' \geq 0$ is already given, and then we get $\bar{c}^T\bar{d} \geq 0 \Rightarrow \bar{c}^T\bar{y} \geq \bar{c}^T\bar{x}$.

We have $d_i = y_i - x_i \ \forall\, i \in N$. But $x_i = 0 \ \forall\, i \in N$ (as $\bar{x}$ is a bfs).

Also, $y_i \geq 0$, as $\bar{y} \in P$ and hence $\bar{y} \geq \bar{0}$.

$\Rightarrow d_i \geq 0 \ \forall\, i \in N. \Rightarrow \bar{c}^T\bar{d} \geq 0$, i.e., $\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y}$.

$\Rightarrow \bar{x}$ is optimal.

Check BT-ILO for proof of statement (b). $\square$

Equivalent definition of optimality conditions

we combine feasibility and optimality

A basis matrix B is optimal if

(a) $B^{-1}\bar{b} \geqslant \bar{0}$, and (feasibity)

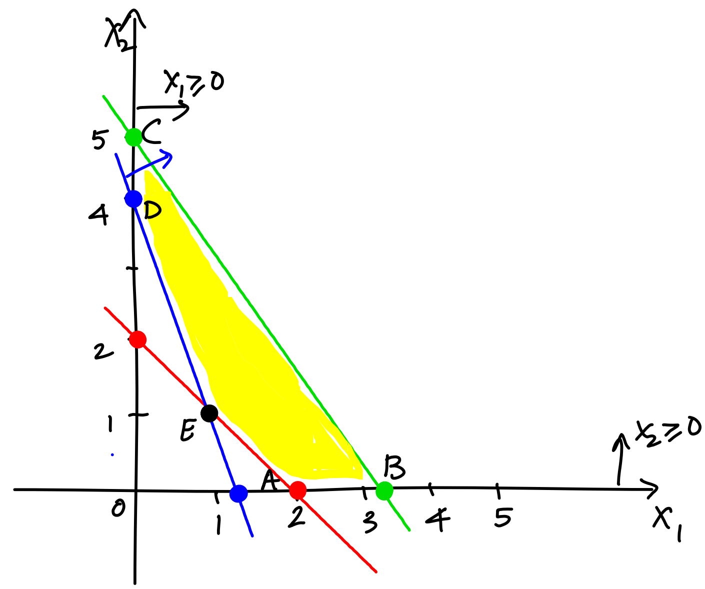
(b) $\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A \geqslant \bar{0}$ (optimality).

At $E(1,1)$, basis is

$\mathcal{B} = \{B(1), B(2), B(3)\} = \{1, 2, 5\}$

$$B = \begin{bmatrix} 1 & 1 & 0 \\ 3 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \qquad B^{-1}\bar{b} = \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix} \geqslant \bar{0}$$

$$\bar{c}' = \begin{bmatrix} 0 \\ 0 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix} \geqslant \bar{0}.$$ So, E is optimal.



We already saw (in Lecture 12) that at $A(2,0)$, the basis is the basis is $\mathcal{B} = \{B(1), B(2), B(3)\} = \{1, 4, 5\}$, giving

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 3 & -1 & 0 \\ 3 & 0 & 1 \end{bmatrix}, \quad B^{-1}\bar{b} = \begin{array}{c} 1 \\ 4 \\ 5 \end{array}\begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} \geqslant \bar{0},$$ but

$$\bar{c}' = \begin{bmatrix} 0 \\ -1 \\ 2 \\ 0 \\ 0 \end{bmatrix} \ngeqslant \bar{0}.$$ Hence A is feasible but not optimal.

Recall: optimality conditions for basis matrix B:

(a) $B^{-1}b \geq \bar{0}$ (feasibility)

(b) $\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A \geq \bar{0}^T$ (optimality).

Simplex Method

How to move to a "better" bfs, given a "good" direction.

Assume all bfs's are nondegenerate. We will deal with degeneracy later.

Let $\bar{x}$ be a bfs. If $\bar{c}' \geq \bar{0}$, $\bar{x}$ is optimal, and we can stop.

If $c_j' < 0$ for some $j \in \mathcal{N}$, then we can move along $\bar{d}$, the j^{th} basic direction, and improve the objective function value. Recall that $\bar{d}$ is given by $\bar{d}_B = -B^{-1}A_j$, $d_j = 1$, $d_i = 0$, $i \in \mathcal{N}, i \neq j$.

If we move θ "units" along $\bar{d}$, x_j increases from 0 to $\theta > 0$. (as $\bar{x} \to \bar{x} + \theta\bar{d}$).

We say that x_j **enters the basis**.

$\mathcal{B} = \{B(1), \ldots, B(m)\}$, and $j \notin \mathcal{B}$ initially.

Subsequently some x_i for $i \in \mathcal{B}$, i.e., a basic variable, has to **leave the basis**, i.e., it becomes 0.

If $B(\ell) = i$, then $x_{B(\ell)} = 0$ in the next bfs.

Recall that the basis should have exactly m indices (out of $1, \ldots, n$). Since j just entered $\mathcal{B}$, one of the current indices should leave.

Which variable leaves?

$\bar{x} \longrightarrow \bar{x}+\theta\bar{d}$. As θ increases, $\bar{c}^T\bar{x}$ decreases. So, go as far as we can go, while staying feasible. So, set $\theta=\theta^*$, where

$$\theta^* = \max\{\theta\geq 0 \mid \bar{x}+\theta\bar{d} \in P\}$$

already satisfies $A(\bar{x}+\theta\bar{d})=\bar{b}$

We need to insure $\bar{x}+\theta\bar{d} \geq \bar{0}$ as well.

$x_i \longrightarrow x_i+\theta d_i \geq 0$:

For $i \in \mathcal{N}$, $x_j=\theta>0$, $x_i=0 \ \forall i \in \mathcal{N}, i\neq j$.

We need to make sure $x_i \geq 0 \ \forall i \in \mathcal{B}$. If $d_i \geq 0$, then $x_i+\theta d_i$ stays >0 (as $x_i>0$ to start with, and $\theta>0$).

So we look at the cases where $d_i<0$, for $i \in \mathcal{B}$. But overall,

$$x_i+\theta d_i \geq 0 \quad \Rightarrow \quad \theta \geq \frac{-x_i}{d_i}, \ i \in \mathcal{B}.$$

If $d_i>0$ for some i, we get $\theta \geq$ (negative #), which is redundant.

(1) If $\bar{d}\geq\bar{0}$, then $\theta\rightarrow\infty$, and $\bar{c}^T\bar{x}\rightarrow-\infty$, i.e., the LP is unbounded (Case III).

(2) If $d_i<0$ for at least one i, then set

$$\theta^* = \min_{\{i \mid d_i<0\}} \left(\frac{-x_i}{d_i}\right)$$

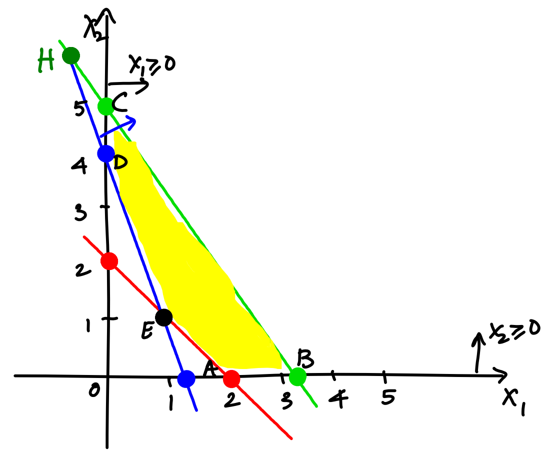
this is termed the minimum-ratio or **min-ratio test**

$$\theta^* = \min_{i\in\mathcal{B} \mid d_{B(i)}<0} \left(\frac{-x_{B(i)}}{d_{B(i)}}\right)$$

it is important to consider the ratio $\frac{-x_i}{d_i}$ only when $d_i<0$ here.

Back to 2D Example

$\mathcal{B} = \{1,3,4\}, \ \mathcal{N} = \{2,5\}.$



At $B(10/3, 0)$, basis has $B(1)=1$, $B(2)=3$, $B(3)=4$.

$$\bar{c}' = \begin{bmatrix} 0 \\ -1/3 \\ 0 \\ 0 \\ -2/3 \end{bmatrix}, \text{ showing that the bfs } \bar{x} = \begin{bmatrix} 10/3 \\ 0 \\ 4/3 \\ 6 \\ 0 \end{bmatrix}$$

is not optimal. For $j=2$, the second basic direction is given by

$$\bar{d} = \begin{bmatrix} -2/3 \\ 1 \\ 1/3 \\ -1 \\ 0 \end{bmatrix}.$$

$d_i < 0$

Hence $\theta^* = \min\left\{ -\frac{10/3}{-2/3}, -\frac{6}{-1} \right\}$
$= \min\{5, 6\} = 5.$

We move $\theta^* = 5$ units along $\bar{d}$ to get to the bfs corresponding to $C(0,5)$.

$$\bar{x} + \theta^* \bar{d} = \begin{bmatrix} 10/3 \\ 0 \\ 4/3 \\ 6 \\ 0 \end{bmatrix} + 5 \begin{bmatrix} -2/3 \\ 1 \\ 1/3 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 3 \\ 1 \\ 0 \end{bmatrix}.$$

point $C(0,5)$.

MATH 464 – Lecture 14 (02/23/2023)

Today: * an iteration of the simplex method
* proofs related to simplex method

An iteration of the Simplex Algorithms

We will describe how to find a starting bfs later on.

1. Start with $\mathcal{B} = \{B(1), \ldots, B(m)\}$ and bfs $\bar{x}$.

2. Find $c_j' = c_j - \bar{c}_B^T B^{-1} A_j$ for $j \in \mathcal{N}$.
   if $c_j' \geq 0 \ \forall j \in \mathcal{N}$ then STOP; $\bar{x}$ is optimal
   else choose a j with $c_j' < 0$. → more on this choice later on!

3. Find $\bar{d}_B = -B^{-1} A_j$
   if $\bar{d}_B \geq \bar{0}$, then $\theta^* = \infty$, optimal cost $= -\infty$; STOP.
   LP is unbounded.

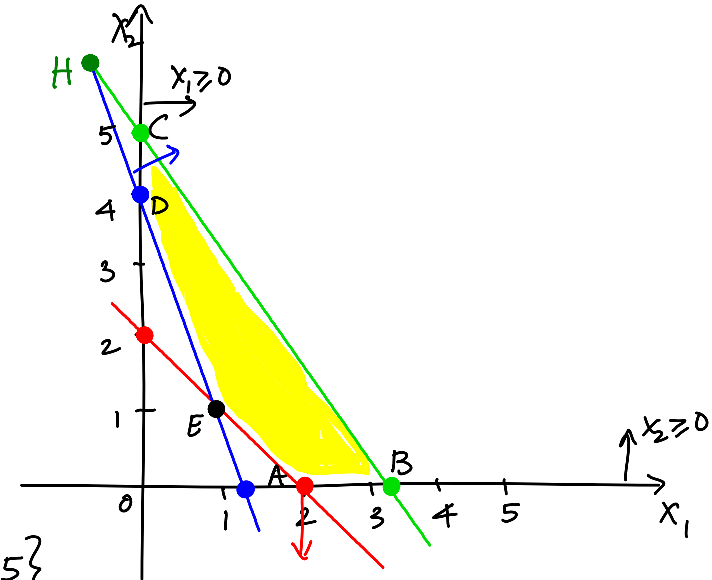
4. Some entry in $\bar{d}_B$ is < 0. Let
$$\theta^* = \min_{i \in \mathcal{B},\, d_{B(i)} < 0} \left(\frac{-x_{B(i)}}{d_{B(i)}} \right) \text{ and let } \theta^* = \frac{-x_{B(\ell)}}{d_{B(\ell)}}$$
   where ℓ: index of the winner of the min-ratio test.

5. Replace $B(\ell)$ in $\mathcal{B}$ with j; → x_j enters the basis
   New basis $\mathcal{B}'$ is given by $B'(i) = \begin{cases} B(i), & i \neq \ell \\ j, & i = \ell \end{cases}$.

   New bfs $\bar{x}' = \bar{x} + \theta^* \bar{d}$ has $x_j' = \theta^*$, and
$$x'_{B(i)} = x_{B(i)} + \theta^* d_{B(i)}, \quad i \in \mathcal{B}, \ i \neq \ell;$$
$$x'_{B(\ell)} = 0.$$

We will prove $\bar{x}'$ is indeed a bfs!

Illustration



Iteration 1

Let's start at $B(10/3, 0)$.

$$A = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 3 & 1 & 0 & -1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 2 \\ 4 \\ 10 \end{bmatrix}$$

$$\bar{c}^T = [2 \; 1 \; 0 \; 0 \; 0]$$

$$\bar{x} = \begin{bmatrix} 10/3 \\ 0 \\ 4/3 \\ 6 \\ 0 \end{bmatrix} \quad \mathcal{B} = \{1,3,4\}, \; \mathcal{N} = \{2,5\}$$

$$\bar{c}'_{\mathcal{N}} = \begin{array}{c} 2 \\ 5 \end{array} \begin{bmatrix} -1/3 \\ -2/3 \end{bmatrix}$$

Let's pick $j=2$: $\bar{d}_B = \begin{array}{c} 1 \\ 3 \\ 4 \end{array} \begin{bmatrix} -2/3 \\ 1/3 \\ -1 \end{bmatrix}$ $\qquad \theta^* = \min\left\{ \underbrace{-\frac{10/3}{-2/3}}_{\ell=1}, -\frac{6}{-1} \right\} = 5.$

$$\bar{x}' = \bar{x} + \theta^* \bar{d} = \begin{bmatrix} 0 \\ 5 \\ 3 \\ 1 \\ 0 \end{bmatrix} \longrightarrow C(0,5).$$

x_2 entered the basis, and x_1 left the basis. $\mathcal{B}' = \{2,3,4\}$.

Iteration 2

$\mathcal{B} = \{2,3,4\}$, $\mathcal{N} = \{1,5\}$, $\bar{x} = \begin{array}{c} \\ 2 \\ 3 \\ 4 \\ \\ \end{array} \begin{bmatrix} 0 \\ 5 \\ 3 \\ 1 \\ 0 \end{bmatrix}$

Reduced costs $\bar{c}' = \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 0 \\ -1/2 \end{bmatrix}$. $\quad j=5$ enters the basis now.

Notice that $x_5 = 0$ at $C(0,5)$; indeed $3x_1 + 2x_2 \leq 10$ is active at C.

$$\bar{d}_B = -B^{-1}A_2 = \begin{array}{c} 2 \\ 3 \\ 4 \end{array}\begin{bmatrix} -1/2 \\ -1/2 \\ -1/2 \end{bmatrix} \qquad \theta^* = \min\left\{\frac{-5}{-1/2}, \frac{-3}{-1/2}, \frac{-1}{-1/2}\right\} = 2$$

$\ell = 3$ here.

So, $B(\ell) = 4$ leaves the basis, as $j=5$ enters.

$$\bar{x}' = \bar{x} + \theta^* \bar{d} = \begin{bmatrix} 0 \\ 4 \\ 2 \\ 0 \\ 2 \end{bmatrix}$$ corresponds to $D(0,4)$

x_4 leaves the basis, x_5 enters; $\mathcal{B}' = \{2, 3, 5\}$.

Iteration 3 $\mathcal{B} = \{2, 3, 5\}$, $\mathcal{N} = \{1, 4\}$, $\bar{x} = \begin{array}{c} \\ 2 \\ 3 \\ \\ 5 \end{array}\begin{bmatrix} 0 \\ 4 \\ 2 \\ 0 \\ 2 \end{bmatrix}$. We get

$$\bar{c}' = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \qquad j=1 \text{ enters the basis.}$$

$$\bar{d}_B = -B^{-1}A_1 = \begin{array}{c} 2 \\ 3 \\ 5 \end{array}\begin{bmatrix} -3 \\ -2 \\ 3 \end{bmatrix} \qquad \theta^* = \min\left\{\frac{-4}{-3}, \frac{-2}{-2}\right\} = 1, \ \ell = 2$$

$B(\ell) = 3$ leaves the basis. $\mathcal{B}' = \{2, 1, 5\}$, $\bar{x}' = \bar{x} + \theta^* \bar{d} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 5 \end{bmatrix}$ corresponds to $E(1,1)$

Iteration 4 $\mathcal{B} = \{2, 1, 5\}$, $\mathcal{N} = \{3, 4\}$, $\bar{x} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 5 \end{bmatrix}$.

$$\bar{c}' = \begin{bmatrix} 0 \\ 0 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix} \geqslant \bar{0}.$$ Hence $\bar{x}$ is an optimal bfs.

Optimal cost $z^* = \bar{c}^T \bar{x} = 3$.

Correctness of the Simplex Method

We present some theorems that certify the correctness of the simplex method. We first show that the algorithm will terminate, and then present results certifying the structure of each solution visited along the way, i.e., that they're adjacent bfs's.

Q. Will the simplex method always terminate?

(BT-ILO Theorem 3.3) Let $P \neq \emptyset$, every bfs be nondegenerate. Then the simplex method terminates after a finite number of iterations with one of two outcomes:

(a) We get an optimal basis $\mathcal{B}$ and an optimal bfs $\bar{x}$; or

(b) We get a direction $\bar{d}$ such that $A\bar{d} = \bar{0}$, $\bar{d} \geq \bar{0}$, and $\bar{c}^T \bar{d} < 0$, showing that the optimal cost is $-\infty$, i.e., the LP is unbounded.

Proof

STOP in Steps 2 and 3 give the two outcomes.

→ nondegeneracy assumption

In each iteration, $\theta^* > 0$, $\bar{c}^T \bar{d} < 0$, so the cost strictly decreases in each step. Hence we do not visit a bfs more than once. But there are only finitely many bfs's — at most $\binom{n}{m}$. Hence the algorithm must terminate after a finite number of steps.

Q. In Step 5, we set $\bar{x}' = \bar{x} + \theta^* \bar{d}$. How can we be sure that $\bar{x}'$ is indeed a bfs?

BT-ILO Theorem 3.2

(a) The columns $A_{B(i)}$, $i \neq l$, and A_j are LI, and hence B', the new basis matrix, is indeed a basis matrix.

(b) $\bar{x}' = \bar{x} + \theta^* \bar{d}$ is a bfs associated with B'.

Proof

We have
$$B' = \begin{bmatrix} | & | & & | & & | & & | \\ A_{B(1)} & A_{B(2)} & \cdots & A_{B(l-1)} & A_j & A_{B(l+1)} & \cdots & A_{B(m)} \\ | & | & & | & & | & & | \end{bmatrix}$$

and
$$B = \begin{bmatrix} | & | & & | & & | & & | \\ A_{B(1)} & A_{B(2)} & \cdots & A_{B(l-1)} & A_{B(l)} & A_{B(l+1)} & \cdots & A_{B(m)} \\ | & | & & | & & | & & | \end{bmatrix}.$$

We want to show columns of B' are LI. Instead, we look at the columns of $B^{-1}B'$. Notice that $B^{-1}B = I$. Hence

$$B^{-1}B' = \left[\, \bar{e}_1 \ \bar{e}_2 \ \cdots \ \bar{e}_{l-1} \ B^{-1}A_j \ \bar{e}_{l+1} \cdots e_m \right], \quad \text{where}$$

$\bar{e}_i$ is the i^{th} unit m-vector. l^{th} entry.

But $B^{-1}A_j = -\bar{d}_B$. We note that $d_{B(l)} \neq 0$, as l is the index of the winner of the min-ratio test. Hence $-\bar{d}_B$ is LI from the remaining $\bar{e}_i$ vectors, which are LI by themselves. Hence the columns of $B^{-1}B'$, and hence that of B', are LI.

(b) We have $A\bar{x}' = \bar{b}$ and $\bar{x}' \geq \bar{0}$ (feasibility), and $\bar{x}'$ is associated with the matrix B' which is a basis matrix. Hence $\bar{x}'$ is a bfs. $\square$

MATH 464 – Lecture 15 (02/28/2023)

Today: * details of simplex method
* revised simplex method

Details of the Simplex Method

Pivot Selection

Choices in Steps 2 and 5: choose a j s.t. $c_j' < 0$ and $\ell = \underset{\{i \in \mathcal{B} \,|\, d_{B(i)} < 0\}}{\arg\min} \left(\frac{-x_{B(i)}}{d_{B(i)}} \right)$

Here are a few options.

(a) Pick j such that $c_j' < 0$ and c_j' is the most negative reduced cost among $j \in \mathcal{N}$ (fastest rate of decrease of $\bar{c}^T \bar{x}$).

(b) Pick j such that $c_j' < 0$ and $|c_j'| \theta^*$ is largest.
(Largest net decrease in $\bar{c}^T \bar{x}$).

Option (a) picks the direction of steepest descent, but we might not be able to move too far in the steepest direction.

Option (b) has more foresight than option (a), but also requires more computation. In (b), we have to find the basic direction ($\bar{d}_B = -B^{-1} A_j$) for each j, and then do the min-ratio test for each j (to find the θ^*).

(c) For really large problems, calculate c_j's one at a time until a $c_j' < 0$ is found, then just go with that j.

We cannot afford to compute the entire $\bar{c}'$ vector — so, just compute c_j' one at a time.

A naive implementation

(one iteration)

1. Given a bfs $\bar{x}$, basis $\mathcal{B} = \{B(1), \dots, B(m)\}$, basis matrix B, solve $\bar{p}^T B = \bar{c}_B^T$ to get $\bar{p}^T = \bar{c}_B^T B^{-1}$, the **simplex multipliers**.

   → same as $B^T \bar{p} = \bar{c}_B$; but we use $\bar{p}^T$ directly afterward.

2. Find reduced costs for non-basic variables

$$\bar{c}_{\mathcal{N}}^{\prime T} = \bar{c}_{\mathcal{N}}^T - \underbrace{\bar{c}_B^T B^{-1}}_{\bar{p}^T} A_{\mathcal{N}} = \bar{c}_{\mathcal{N}}^T - \bar{p}^T A_{\mathcal{N}}.$$

$\mathcal{N}$ – nonbasic indices

If $\bar{c}_{\mathcal{N}}^{\prime T} \geq \bar{0}^T$ current bfs is optimal; STOP.

Else pick a j such that $c_j' < 0$.

systems of linear equations!

3. Solve $B \bar{d}_B = -A_j$ to get $\bar{d}_B = -B^{-1} A_j$. ← j^{th} basic direction

If $\bar{d}_B \geq \bar{0}$ STOP. $\bar{c}^T \bar{x} \to -\infty$

else find

$$\theta^* = \min_{\{i \in \mathcal{B} \mid d_{B(i)} < 0\}} \left(\frac{-x_{B(i)}}{d_{B(i)}} \right) = \left(\frac{-x_{B(\ell)}}{d_{B(\ell)}} \right).$$

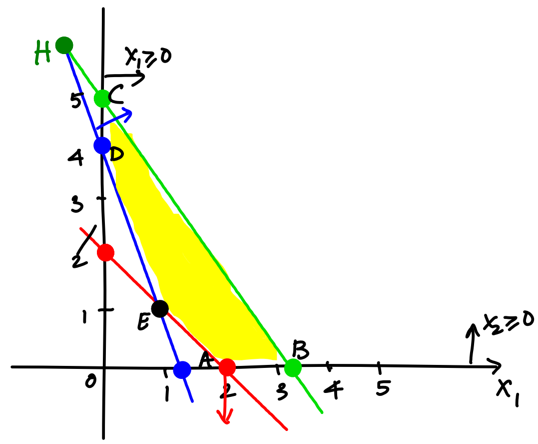
4. Change basis: replace $B(\ell)$ with j, update bfs to $\bar{x}'$ such that $x_j' = \theta^*$, $x_{B(i)}' = x_{B(i)} + \theta^* d_{B(i)}$ $\forall i \neq j, i \in \mathcal{B}$.

Solving the two systems of linear equations repeatedly could also prove costly! Hence the "naive" implementation.

Revised Simplex Method

From one iteration to the next, B changes by only one column. If we know B^{-1} from previous iteration, we could use it to find new B^{-1} (or $(B')^{-1}$) quickly. So we store B^{-1}, and update it efficiently in each step.

$$
\begin{array}{lrl}
\min & 2x_1+x_2 & \\
\text{s.t.} & x_1+x_2-x_3 & = 2 \\
& 3x_1+x_2 \quad -x_4 & = 4 \\
& 3x_1+2x_2 \quad +x_5 & = 10 \\
& x_j \geq 0 \ \forall j &
\end{array}
$$



Correspondences between bfs's and corner points

pt	Basis $\mathcal{B}$: $B(1)$ $B(2)$ $B(3)$	B	B^{-1}	x_1	x_2	x_3	x_4	x_5
A	1 4 5	$\begin{bmatrix}1&0&0\\3&-1&0\\3&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\3&-1&0\\-3&0&1\end{bmatrix}$	2	0	0	2	4
B	1 3 4	$\begin{bmatrix}1&-1&0\\3&0&-1\\3&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1/3\\-1&0&1/3\\0&-1&1\end{bmatrix}$	10/3	0	4/3	6	0
C	2 3 4	$\begin{bmatrix}1&-1&0\\1&0&-1\\2&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1/2\\-1&0&1/2\\0&-1&1/2\end{bmatrix}$	0	5	3	1	0
D	2 3 5	$\begin{bmatrix}1&-1&0\\1&0&0\\2&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&0\\-1&1&0\\0&-2&1\end{bmatrix}$	0	4	2	0	2
E	1 2 5	$\begin{bmatrix}1&1&0\\3&1&0\\3&2&1\end{bmatrix}$	$\begin{bmatrix}-1/2&1/2&0\\3/2&-1/2&0\\-3/2&-1/2&1\end{bmatrix}$	1	1	0	0	5

From one iteration to next, the basis matrix B changes by only one column. So, if we know B^{-1} from the previous iteration, we can use it to find the new B^{-1} efficiently.

Q. Given B^{-1}, how do we find new B^{-1} (or $(B')^{-1}$)?

We know $B^{-1}B = I$, i.e., $\qquad \bar{e}_i$: i^{th} unit m-vector

$$B^{-1}\left[A_{B(1)}\ A_{B(2)} \cdots A_{B(\ell)} \cdots A_{B(m)}\right] = \left[\bar{e}_1\ \bar{e}_2 \cdots \bar{e}_\ell \cdots \bar{e}_m\right]$$

New basis matrix $\quad B' = \left[A_{B(1)} \cdots A_j \cdots A_{B(m)}\right]$ $\qquad x_j$ entering the basis; replacing $x_{B(\ell)}$; $\{B(\ell) \to j\}$ in B

$$\Longrightarrow B^{-1}B' = \left[\bar{e}_1\ \bar{e}_2 \cdots B^{-1}A_j \cdots \bar{e}_m\right]; \text{ recall } \bar{d}_B = -B^{-1}A_j.$$

$$\Longrightarrow B^{-1}B' = \left[\bar{e}_1\ \bar{e}_2 \cdots -\bar{d}_B \cdots \bar{e}_m\right].$$

Recall EROs to invert an $n \times n$ matrix $[A\,|\,I] \xrightarrow{\text{EROs}} [I\,|\,A^{-1}]$

Following this idea, we need a few more EROs to convert $-\bar{d}_B$ to $\bar{e}_\ell$.

Let $Q B^{-1}B' = I$, where Q represents the EROs on $B^{-1}B'$ that convert it to I.

$B^{-1}B'$ is almost I, except of $-\bar{d}_B$ in the ℓ^{th} column. Q represents the EROs that convert $-\bar{d}_B$ to $\bar{e}_\ell$.

$$\Longrightarrow Q B^{-1} = (B')^{-1}$$

We can start with $[B^{-1} | -\bar{d}_B]$, and do EROs to convert $-\bar{d}_B$ to $\bar{e}_\ell$. We will get $(B')^{-1}$ sitting in place of B^{-1}.

Back to the example

When we go from $B(10/3, 0)$ to $C(0,5)$, the basis matrix sees only one column changed:

$$\begin{bmatrix} 1 & -1 & 0 \\ 3 & 0 & -1 \\ 3 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 2 & 0 & 0 \end{bmatrix}$$

$$\mathcal{B} = \{\overset{\ell}{1}, 3, 4\}$$

$$\mathcal{B}' = \{\underset{j}{2}, 3, 4\}$$

We have $j=2$ (entering) and $\ell=1$ (leaving).

At $B(10/3, 0)$, $B^{-1} = \begin{bmatrix} 0 & 0 & 1/3 \\ -1 & 0 & 1/3 \\ 0 & -1 & 1 \end{bmatrix}$. $\Rightarrow \bar{d}_B = -B^{-1}A_j$

$$\Rightarrow -\bar{d}_B = B^{-1}A_j = \begin{bmatrix} 0 & 0 & 1/3 \\ -1 & 0 & 1/3 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2/3 \\ -1/3 \\ 1 \end{bmatrix}.$$

$$[B^{-1} | -\bar{d}_B] \xrightarrow{\text{EROs}} [(B')^{-1} | \bar{e}_\ell] \qquad \ell = 1 \text{ here.}$$

$$\left[\begin{array}{ccc|c} 0 & 0 & 1/3 & 2/3 \\ -1 & 0 & 1/3 & -1/3 \\ 0 & -1 & 1 & 1 \end{array}\right] \xrightarrow[\substack{\text{and then} \\ R_2 + \frac{1}{3}R_1 \\ R_3 - R_1}]{R_1 \times 3/2} \left[\begin{array}{ccc|c} 0 & 0 & 1/2 & 1 \\ -1 & 0 & 1/2 & 0 \\ 0 & -1 & 1/2 & 0 \end{array}\right] \Rightarrow (B')^{-1} = \underbrace{\begin{bmatrix} 0 & 0 & 1/2 \\ -1 & 0 & 1/2 \\ 0 & -1 & 1/2 \end{bmatrix}}$$

this is indeed the $(B')^{-1}$ at the bfs corresponding to $C(0,5)$.

If we were to invert B' from scratch, we would be solving m such systems, and not just one!

One iteration of the revised Simplex method

1. Start with basis $\mathcal{B} = \{B(1), \dots, B(m)\}$, basis matrix B, bfs $\bar{x}$.
   $\bar{x} = B^{-1}\bar{b}$; we assume B^{-1} is known.

2. Find $\bar{c}_{N}'^{T} = \bar{c}_{N}^{T} - \bar{c}_B^T B^{-1} A_{N}$.

3. If $\bar{c}_{N}'^{T} \geq \bar{0}$, then current bfs is optimal. STOP.
   Else pick non-basic x_j to enter the basis (with $c_j' < 0$).

4. Find $\bar{d}_B = -B^{-1}A_j$.
   If $\bar{d}_B \geq \bar{0}$, STOP. LP is unbounded, $\bar{c}^T\bar{x} \to -\infty$.

5. Choose $x_{B(\ell)}$ to leave the current basis such that

$$\left(\frac{-x_{B(\ell)}}{d_{B(\ell)}}\right) = \theta^* = \min_{\{d_{B(i)}<0,\ i \in \mathcal{B}\}} \left(\frac{-x_{B(i)}}{d_{B(i)}}\right).$$

   New bfs is $\bar{x}' = \bar{x} + \theta^*\bar{d}$.

6. Find new B^{-1} (or $(B')^{-1}$) using EROs:

$$\left[B^{-1} \,\middle|\, -\bar{d}_B\right] \xrightarrow{\text{EROs}} \left[(B')^{-1} \,\middle|\, \bar{e}_\ell\right].$$

MATH464 – Lecture 16 (03/02/2023)

Today: * Example for revised simplex method
* full tableau implementation

Problems from Hw6

5. **Exercise 2.16** Consider the set $S = \{\mathbf{x} \in \Re^n \mid x_1 = \cdots = x_{n-1} = 0,\ 0 \le x_n \le 1\}$. Could this be the feasible set of a problem in standard form?

P in standard form is $\{\bar{x} \in \mathbb{R}^n \mid A\bar{x} = \bar{b},\ \bar{x} \geqslant \bar{0}\}$, with $\operatorname{rank}(A) = m$, $m \le n$.

The given set (let's call it S) is obviously not in standard form. The question is whether this set could describe a set that is in standard form.

You should assume $n \geqslant 2$.

Note that $\bar{x} = \bar{0} \in S$. So, if $S = P$, then $\bar{0} \in P$ as well.

$\Rightarrow A(\bar{0}) = \bar{b} \Rightarrow \bar{b} = \bar{0}$ in the standard form.

with $\bar{b} = \bar{0}$, if $\bar{x} \in P$, $\lambda\bar{x} \in P \ \forall \lambda \geqslant 0$

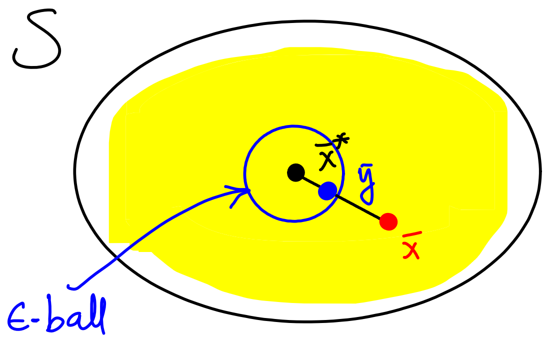
$$A(\lambda\bar{x}) = \lambda A(\bar{x}) = \lambda(\bar{b}) = \lambda\bar{0} = \bar{0}. \quad \& \quad \lambda\bar{x} \geqslant \bar{0}.$$

So $\lambda > 1$ is also good.

But $\lambda\bar{x}_n \not\leq 1$ for $\lambda > 1$.

So $\lambda\bar{x} \notin S$ for $\lambda > 1$, giving an obstruction for $S = P$.

6. Problem 3.6. Show that a local minimum $\bar{x}^*$ of a convex function over a convex set S is the global minimum.



$\bar{x}^*$ is a local optimum of $f \Rightarrow$

$\exists\ \epsilon > 0$, s.t.

$f(\bar{x}^*) \leq f(\bar{x})\ \forall \bar{x}$ s.t. $\|\bar{x} - \bar{x}^*\| \leq \epsilon$

Need to prove $\bar{x}^*$ is global minimum, i.e., $f(\bar{x}^*) \leq f(\bar{x})\ \forall \bar{x} \in S$ (and not just inside the ϵ-ball around $\bar{x}^*$).

Proof by contradiction: Assume $\exists\ \bar{x}$ outside the ϵ-ball centered at $\bar{x}^*$ with $f(\bar{x}) < f(\bar{x}^*)$, i.e., $\bar{x}^*$ is **not** a global minimum.

Consider a point $\bar{y}$ on the line segment connecting $\bar{x}^*$ and $\bar{x}$ that is within the ϵ-ball @ $\bar{x}^*$, i.e., $\|\bar{y} - \bar{x}^*\| < \epsilon$.

$$\Rightarrow f(\bar{x}^*) \leq f(\bar{y}) \quad \text{——} (1)$$

Q. Why are guaranteed to find such a $\bar{y}$?

$\bar{y}$ is a point on the line segment joining $\bar{x}^*$ and $\bar{x}$

$\Rightarrow \bar{y} = \lambda \bar{x}^* + (1-\lambda)\bar{x}$ for some $\lambda \in (0,1)$.

Q. Why do we choose λ strictly between 0 and 1?

$$\Rightarrow f(\bar{y}) = f(\lambda \bar{x}^* + (1-\lambda)\bar{x}) \leq \lambda f(\bar{x}^*) + (1-\lambda)\bar{x} \quad \text{——} (2)$$

why?

Combine (1) & (2) to get a contradiction...

Hw7: AMPL problems. You **must** separate the model from data. A good rule to follow: do **not** include any actual numbers, i.e., data in your model file.

An example for the Revised Simplex Method

Recall how we maintain and update B^{-1}:

$$\left[\, B^{-1} \,\middle|\, -d_B \,\right] \xrightarrow{\text{EROs}} \left[\, (B')^{-1} \,\middle|\, \bar{e}_\ell \,\right]$$

Solve using the revised simplex method.

$$\left.\begin{array}{ll} \min & -x_1 - x_2 \\ \text{s.t.} & 2x_1 + x_2 \le 4 \\ & 3x_1 + 5x_2 \le 15 \\ & x_1, x_2 \ge 0 \end{array}\right\} \qquad \begin{array}{ll} \min & -x_1 - x_2 \\ \text{s.t.} & 2x_1 + x_2 + x_3 = 4 \\ & 3x_1 + 5x_2 + x_4 = 15 \\ & x_j \ge 0 \;\forall j \end{array} \left.\right\} \qquad \begin{array}{l} \bar{c}^T = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 \end{bmatrix} \end{array} \\ A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 3 & 5 & 0 & 1 \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 4 \\ 15 \end{bmatrix}. \\ m = 2,\ n = 4 \end{array}$$

Iteration 1

$B(1) = 3,\ B(2) = 4 \quad \mathcal{B} = \{3,4\},\ \mathcal{N} = \{1,2\}. \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ B^{-1} = B = I.$

$$\bar{c}_B^T = [0 \ \ 0]. \qquad \bar{c}_{\mathcal{N}}'^{T} = \bar{c}_{\mathcal{N}}^T - \underbrace{\cancel{\bar{c}_B^T}}_{=0} B^{-1} N = \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} -1 & -1 \end{bmatrix} \end{array} \quad j = 1 \quad (x_1 \text{ enters}).$$

$$\bar{x}_B = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = B^{-1}\bar{b} = \begin{bmatrix} 4 \\ 15 \end{bmatrix}. \qquad \bar{d}_B = -B^{-1}A_1 = -\begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

$$\theta^* = \min\left\{ \frac{-4}{-2}\checkmark,\ \frac{-15}{-3} \right\} = 2 \qquad \ell = 1 \qquad B(1) = 3 \text{ leaves the basis } (x_3 \text{ leaves}).$$

$$\mathcal{B}' = \{1,4\}. \qquad \bar{x}' = \bar{x} + \theta^* \bar{d} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 15 \end{bmatrix} + 2\begin{bmatrix} 1 \\ 0 \\ -2 \\ -3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 9 \end{bmatrix}.$$

$$\left[\, B^{-1} \,\middle|\, -\bar{d}_B \,\right] \xrightarrow{\text{EROs}} \left[\, (B')^{-1} \,\middle|\, \bar{e}_\ell \,\right] \qquad \bar{e}_\ell = \bar{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ here.}$$

$$\left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 3 \end{array}\right] \xrightarrow[\substack{\text{then} \\ R_3 - 3R_1}]{R_1 \times \frac{1}{2}} \left[\begin{array}{cc|c} \frac{1}{2} & 0 & 1 \\ -\frac{3}{2} & 1 & 0 \end{array}\right] \implies \text{new } B^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ -\frac{3}{2} & 1 \end{bmatrix}.$$

Iteration 2

$\mathcal{B} = \{1,4\}$, $\mathcal{N} = \{2,3\}$. $\quad \bar{c}_B^T = [-1\ \ 0]$

$$\bar{c}_{\mathcal{N}}'^T = \bar{c}_{\mathcal{N}}^T - \bar{c}_B^T B^{-1} N = \begin{array}{cc} 2 & 3 \\ {[-1} & {0]} \end{array} - \underbrace{\underbrace{[-1\ \ 0]\begin{bmatrix} 1/2 & 0 \\ -3/2 & 1 \end{bmatrix}}_{[-1/2\ \ 0]}\begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix}}_{[-1/2\ \ -1/2]}$$

$$= \begin{array}{cc} 2 & 3 \\ {[-1/2} & {1/2]} \end{array} \qquad j = 2 \ (x_2 \text{ enters}).$$

no tie to break here!

$$\bar{d}_B = -B^{-1}A_2 = -\begin{bmatrix} 1/2 & 0 \\ -3/2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{array}{c} 1 \\ 4 \end{array}\begin{bmatrix} -1/2 \\ -7/2 \end{bmatrix} \qquad \bar{x} = \begin{array}{c} 1 \\ \\ \\ 4 \end{array}\begin{bmatrix} 2 \\ 0 \\ 0 \\ 9 \end{bmatrix}$$

$$\theta^* = \min\left\{\frac{-2}{-1/2}, \frac{-9}{-7/2}\right\} = \frac{18}{7}, \quad l = 2 \text{ here. } \mathcal{B}(2) = 4 \text{ leaves.}$$

$\mathcal{B}' = \{1,2\}$ (new basis). $\quad \bar{x}' = \bar{x} + \theta^* \bar{d} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 9 \end{bmatrix} + \frac{18}{7}\begin{bmatrix} -1/2 \\ 1 \\ 0 \\ -7/2 \end{bmatrix} = \begin{bmatrix} 5/7 \\ 18/7 \\ 0 \\ 0 \end{bmatrix}.$

Update B^{-1}:

$$\left[\begin{array}{cc|c} 1/2 & 0 & 1/2 \\ -3/2 & 1 & 7/2 \end{array}\right] \xrightarrow[\text{then } R_1 - \frac{1}{2}R_2]{R_2 \times 2/7} \left[\begin{array}{cc|c} 5/7 & -1/7 & 0 \\ -3/7 & 2/7 & 1 \end{array}\right] \qquad \text{New } B^{-1} = \begin{bmatrix} 5/7 & -1/7 \\ -3/7 & 2/7 \end{bmatrix}.$$

Iteration 3

$\mathcal{B} = \{1,2\}$, $\mathcal{N} = \{3,4\}$. $\quad \bar{c}_B^T = \begin{array}{cc} 1 & 2 \\ {[-1} & {-1]} \end{array} \quad \bar{c}_{\mathcal{N}}^T = \begin{array}{cc} 3 & 4 \\ {[0} & {0]} \end{array}$

$$\bar{c}_{\mathcal{N}}'^T = \bar{c}_{\mathcal{N}}^T - \bar{c}_B^T B^{-1} N = [0\ \ 0] - [-1\ \ -1]\begin{bmatrix} 5/7 & -1/7 \\ -3/7 & 2/7 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= [2/7\ \ 1/7] \geq 0$$

$\bar{x} = \begin{bmatrix} 5/7 \\ 18/7 \\ 0 \\ 0 \end{bmatrix}$ is an optimal bfs, with the optimal objective function value $z^* = \bar{c}^T \bar{x} = \dfrac{-23}{7}$.

Full-Tableau Implementation (LP in standard form)

For large LPs, the revised simplex method is the efficient way to solve them. But we will also introduce a compact way of representing the numbers, and implementing all the computations – the tableau simplex method.

We maintain and update $[\bar x_B \mid B^{-1}A]$ or $B^{-1}[\bar b \mid A]$. This is an $m \times (n+1)$ matrix, or table, called the **simplex tableau**.

$$\begin{array}{cccccc} 0 & 1 & 2 & \cdots & n \\ \end{array}$$
$$\left[B^{-1}\bar b \mid B^{-1}A_1 \;\; B^{-1}A_2 \cdots B^{-1}A_n \right]$$

Nothing new here!

$B^{-1}(A\bar x = \bar b) \Rightarrow$

$(B^{-1}A)\bar x = B^{-1}\bar b$

$[B^{-1}\bar b \mid B^{-1}A]$

0^{th} column: values of the basic variables

i^{th} column: has $B^{-1}A_i$, for $i = 1, \ldots, n$.

If x_j is entering, the column having $B^{-1}A_j = -\bar d_B$ is the **pivot column**.

If $x_{B(\ell)}$ leaves the basis ($1 \le \ell \le n$), then the ℓ^{th} row is the **pivot row**. The (ℓ, j)-element is the **pivot element** or just the pivot.

We include the costs $\bar c^T$ and the objective function $z = \bar c^T \bar x$ at the top as the zero-th row.

$\bar c'$ (reduced cost vector)

$$\begin{array}{c|c|c} & 0 & 1 \;\; 2 \; \cdots \; n \\ \hline 0 & -\bar c_B^T B^{-1}\bar b & \bar c^T - \bar c_B^T B^{-1} A \\ \hline 1, 2, \ldots, m & B^{-1}\bar b & B^{-1}A \end{array}
\;=\;
\begin{array}{c|c|cccccc} & 0 & 1 & 2 & \cdots & j & \cdots & n \\ \hline 0 & -\bar c_B^T B^{-1}\bar b & c_1' & c_2' & \cdots & c_j' & \cdots & c_n' \\ \hline 1 & x_{B(1)} & & & & & & \\ 2 & x_{B(2)} & B^{-1}A_1 & B^{-1}A_2 & \cdots & B^{-1}A_j & \cdots & B^{-1}A_n \\ \vdots & \vdots & & & & & & \\ m & x_{B(m)} & & & & & & \end{array}$$

$-z = -\bar c_B^T \bar x_B$

MATH 464 – Lecture 17 (03/07/2023)

Today: * Hw7 problems
* tableau implementation of simplex

Hw 7

Exercise 3.2 (Optimality conditions) Consider the problem of minimizing $\mathbf{c}'\mathbf{x}$ over a polyhedron P. Prove the following:

(a) A feasible solution $\mathbf{x}$ is optimal if and only if $\mathbf{c}'\mathbf{d} \geq 0$ for every feasible direction $\mathbf{d}$ at $\mathbf{x}$.

A B

(b) A feasible solution $\mathbf{x}$ is the unique optimal solution if and only if $\mathbf{c}'\mathbf{d} > 0$ for every nonzero feasible direction $\mathbf{d}$ at $\mathbf{x}$.

A iff $B \equiv A \Rightarrow B$ and $B \Rightarrow A$

$(\Rightarrow)$ $\bar{x}$ is optimal. $\Rightarrow$ $\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y} \;\; \forall \bar{y} \in P$.

Want to show: $\bar{c}^T\bar{d} \geq 0 \;\; \forall$ feasible direction $\bar{d}$ at $\bar{x}$.

$\bar{d}$ is a feasible direction at $\bar{x} \Rightarrow \exists\, \theta > 0$ s.t. $\bar{x} + \theta\bar{d} \in P$

$\bar{c}^T(\bar{y} - \bar{x}) \geq 0$

Argue that every feasible direction at $\bar{x}$ can be written as $\bar{y} - \bar{x}$.

This could help in the reverse direction below as well!

$(\Leftarrow)$ $\bar{c}^T\bar{d} \geq 0 \;\; \forall$ feasible direction $\bar{d}$ at $\bar{x}$

$\exists\, \theta > 0$ s.t. $\bar{x} + \theta\bar{d} \in P$

from this, derive $\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y}$

try to use $\bar{x} + \theta\bar{d}$ as $\bar{y}$

Exercise 3.3 Let $\mathbf{x}$ be an element of the standard form polyhedron $P = \{\mathbf{x} \in \Re^n \mid \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$. Prove that a vector $\mathbf{d} \in \Re^n$ is a feasible direction at $\mathbf{x}$ if and only if $\mathbf{Ad} = \mathbf{0}$ and $d_i \geq 0$ for every i such that $x_i = 0$.

A

B

$A \Leftrightarrow B$

$(\Rightarrow)$ $\bar{d}$ is a feasible direction @ $\bar{x}$: $\exists \theta > 0$ s.t. $\bar{x} + \theta \bar{d} \in P$

$\Rightarrow A(\bar{x}+\theta\bar{d}) = \bar{b}$, $\bar{x}+\theta\bar{d} \geqslant \bar{0}$

But $A\bar{x} = \bar{b} \Rightarrow \theta A\bar{d} = \bar{0} \Rightarrow A\bar{d} = \bar{0}$.

$\bar{x}+\theta\bar{d} \geqslant \bar{0}$, $x_i = 0 \Rightarrow \theta d_i \geqslant 0$; $\theta > 0 \Rightarrow d_i \geqslant 0$.

$(\Leftarrow)$ Given $A\bar{d} = \bar{0}$, $d_i \geqslant 0$ when $x_i = 0$, show (for $\bar{x} \in P$)

$\exists \theta > 0$ s.t. $A(\bar{x}+\theta\bar{d}) = \bar{b}$ and $\bar{x}+\theta\bar{d} \geqslant \bar{0}$.

We already have $A\bar{x} = \bar{b}$, and now have $A\bar{d} = \bar{0}$.

$\Rightarrow A(\bar{x}+\theta\bar{d}) = \bar{b}$

To show $\exists \theta > 0$ s.t. $\bar{x}+\theta\bar{d} \geqslant \bar{0}$,

think about the min-ratio test!

Exercise 3.6 (Conditions for a unique optimum) Let $\mathbf{x}$ be a basic feasible solution associated with some basis matrix $\mathbf{B}$. Prove the following:

(a) If the reduced cost of every nonbasic variable is positive, then $\mathbf{x}$ is the unique optimal solution.

Use arguments similar to those used for Problem 3.1 (in Homework 6).

Full Tableau Implementation of the Simplex Method

reduced costs $\bar{c}'^T$

	0	1 2 ... n
0	$-\bar{c}_B^T B^{-1} b$	$\bar{c}^T - \bar{c}_B^T B^{-1} A$
1, 2, ..., m	$B^{-1} b$	$B^{-1} A$

$$-z = -\bar{c}_B^T x_B$$

=

		0	1	2	...	j	...	n
0		$-\bar{c}_B^T B^{-1} \bar{b}$	c_1'	c_2'	...	c_j'	...	c_n'
1	$x_{B(1)}$							
2	$x_{B(2)}$		$B^{-1}A_1$	$B^{-1}A_2$	...	$B^{-1}A_j$	...	$B^{-1}A_n$
⋮	⋮							
m	$x_{B(m)}$							

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \le 4 \\ & 3x_1 + 5x_2 \le 15 \\ & x_1, x_2 \ge 0 \end{aligned} \quad \Bigg\} \quad \begin{aligned} \min \quad & -x_1 - x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 + x_3 = 4 \\ & 3x_1 + 5x_2 + x_4 = 15 \\ & x_j \ge 0 \ \forall j \end{aligned} \quad \Bigg\}$$

$$\begin{array}{cccccc} & & 1 & 2 & 3 & 4 \\ \bar{c}^T = & [& -1 & -1 & 0 & 0 \ ] \end{array}$$

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 3 & 5 & 0 & 1 \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 4 \\ 15 \end{bmatrix}.$$

$m = 2, n = 4$

	z	x_1	x_2	x_3	x_4	min-ratio candidates	
	0	-1	-1	0	0		$R_0 + R_1'$
$x_3 =$	4	2	1	1	0	2	$R_1(\frac{1}{2}) = R_1'$
$x_4 =$	15	3	5	0	1	5	$R_2 - 3R_1'$
	2	0	$-\frac{1}{2}$	$\frac{1}{2}$	0		
$x_1 =$	2	1	$\frac{1}{2}$	$\frac{1}{2}$	0	4	
$x_4 =$	9	0	$\frac{7}{2}$	$-\frac{3}{2}$	1	$\frac{18}{7}$	
	$\frac{23}{7}$	0	0	$\frac{2}{7}$	$\frac{1}{7}$		
$x_1 =$	$\frac{5}{7}$	1	0	$\frac{5}{7}$	$-\frac{1}{7}$		
$x_2 =$	$\frac{18}{7}$	0	1	$-\frac{3}{7}$	$\frac{2}{7}$		

B^{-1}

We first scale the pivot row so that the pivot is 1. We use replacement EROs to zero out the rest of the pivot column.

Optimal solution is $x_1 = \frac{5}{7}$, $x_2 = \frac{18}{7}$, with $z^* = -\frac{23}{7}$.

◯ : We indicate the pivot by circling the entry.

Since we maintain $B^{-1}A$, if some columns of A form I, the identity matrix, B^{-1} will be sitting in those columns.

We add the labels of basic variables on the left of each tableau, mainly to improve readability. One could identify the basic variables by spotting the unit vectors among the variable columns.

Why use $-\bar{c}_B^T\bar{x}_B$ in Row-0, Column-0 of the tableau?

If all constraints were $\leq$, we could choose the m slack variables in our starting bfs, and all original $x_j = 0$ to start with.

Hence $\bar{z} = \bar{c}^T\bar{x} = 0$ to start with.

Row-0: $[\,0 \mid \bar{c}^T\,] - \bar{g}^T[\,\bar{b} \mid A\,]$ where $\bar{g}^T = \bar{c}_B^T B^{-1}$ are multipliers.

i.e., a combination of m replacement EROs.

In columns 1 to n, we get $\bar{c}^T - \bar{g}^T A = \bar{c}^T - \bar{c}_B^T B^{-1} A = \bar{c}'$.

In column 0, we get $z \leftarrow z - \bar{g}^T\bar{b} = z - \bar{c}_B^T B^{-1}\bar{b}$

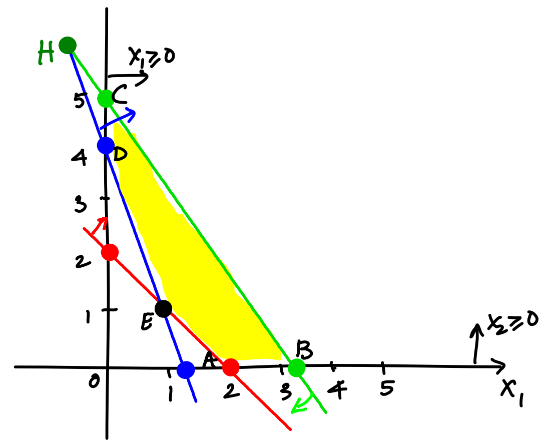
"replace"

$$= z - \bar{c}_B^T\bar{x}_B$$

So, if we start with $z=0$, we get $-\bar{c}_B^T\bar{x}_B$ in Row-0, Column-0.

Back to our favorite example

$$\begin{aligned}
\min \quad & 2x_1 + x_2 \\
\text{s.t.} \quad & x_1 + x_2 - x_3 = 2 \\
& 3x_1 + x_2 - x_4 = 4 \\
& 3x_1 + 2x_2 + x_5 = 10 \\
& x_j \geq 0 \ \forall j
\end{aligned}$$



We start at the bfs $\equiv B(10/3, 0)$, i.e., $\mathcal{B} = \{1, 3, 4\}$.

We will talk about how to find a starting bfs in general later on.

We move from B to A to E.

See the Course web page for the Matlab session. (for the first iteration).

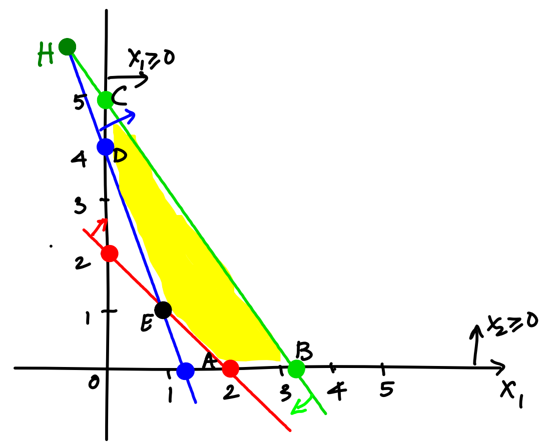
We'll finish the calculations in the next lecture...

MATH464 – Lecture 18 (03/09/2023)

Today: * Examples of tableau simplex
* cycling in tableau simplex

Back to tableau simplex example on our popular LP:

$$\begin{array}{lllll} \min & 2x_1+x_2 & & & \\ \text{s.t.} & x_1+x_2-x_3 & & & = 2 \\ & 3x_1+x_2 & -x_4 & & = 4 \\ & 3x_1+2x_2 & & +x_5 & = 10 \\ & x_j \geqslant 0 \ \forall j & & & \end{array}$$



We start at the bfs $\equiv B(10/3, 0)$, i.e., $\mathcal{B}=\{1,3,4\}$.

We move from B to A to E.

We had done the first iteration (B→A) in the last lecture.
Today, we do the next iteration (A→E).

See the MATLAB session at

https://www.math.wsu.edu/faculty/bkrishna/FilesMath464/S23/Software/Lec18_03092023_Session.txt

Simplex Method and Cycling

BT-ILO Example 3.6 (pg 104)

The book has 3 in place of 0; the value does not change at all, and we get back to this tableau in 6 iterations.

	rhs	x_1	x_2	x_3	x_4	x_5	x_6	x_7
	0	$-3/4$	20	$-1/2$	6	0	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	6	0	0	1

Notice that the bfs is degenerate here — $x_5 = x_6 = 0$ in the bfs!

Pivoting Rules

1. Choose non-basic x_j with most negative c_j' to enter.
2. If tied for leaving variable, choose one with the smallest index.

   e.g., x_2, x_5 are candidates (in Rows 5 & 3, respectively) pick x_2 to leave, as $2 < 5$ (even though it is basic in a row higher up than the row x_5 is basic in).

We get back to the starting tableau after 6 iterations!

See the MATLAB session at

https://www.math.wsu.edu/faculty/bkrishna/FilesMath464/S23/Software/Lec18_03092023_Session.txt

Ways to avoid cycling

1. lexicographic pivoting rule (dictionary order to choose leaving variable).
2. Bland's rule (min index rule).

Lexicographic ordering (dictionary ordering)

Def Let $\bar{u}, \bar{v} \in \mathbb{R}^n$. $\bar{u}$ is **lexicographically larger** (smaller) than $\bar{v}$ if

1. $\bar{u} \neq \bar{v}$, and
2. the first non-zero component of $\bar{u}-\bar{v}$ is positive (negative).

We write $\bar{u} \overset{L}{\succ} \bar{v}$ $(\bar{u} \overset{L}{\prec} \bar{v})$.

Example

$$\overset{\bar{u}}{(1,2,3,4)} \overset{L}{\succ} \overset{\bar{v}}{(1,2,0,7)} \qquad \bar{u}-\bar{v} = (0,0,3,-3)$$

$$(0,5,8,100) \overset{L}{\prec} (1,0,0,0)$$

When $\bar{u} \overset{L}{\succ} \bar{0}$, we say $\bar{u}$ is lexicographically positive (or lex-positive).

e.g., $\bar{u} = (0,0,1,-5,-7,2)$ is lex-positive.

Notice that $\bar{u}$ need not be positive to be lex-positive!

In the lexicographic pivoting rule, we will scale all candidate rows (for leaving variable) so that the entries in the pivot column are all 1's. Then we pick the lexicographically smallest one as ℓ (so $x_{B(\ell)}$ leaves the basis). More on this approach after the break ...

MATH464 – Lecture 19 (03/21/2023)

Today: * lexicographic pivoting rule
* big-M method

Recall: Lexicographic order: $\bar{u} \overset{L}{>} \bar{v}$ if the first nonzero entry of $\bar{u}-\bar{v}$ is positive.

Lexicographic Pivoting Rule (for tableau simplex method)

1. Choice of entering variable is arbitrary (as long as $c_j' < 0$).
2. For each a_{ij} in the pivot column j $(B^{-1}A_j)$ that is > 0, we divide the entire Row-i including column-0, by a_{ij} so that we get a 1 in place of a_{ij}. We then choose the lexicographically smallest row, say Row-ℓ. The corresponding basic variable $x_{B(\ell)}$ leaves the basis.

Back to the cycling example:

Objective function does not change for iterations 1-3, just as before. But it improves in iterations 4 and 5, giving the optimal solution as $z^* = -5/4$ and $x_1 = 1$, $x_3 = 1$, $x_5 = 3/4$.

See the course web page for the Matlab session:
http://www.math.wsu.edu/faculty/bkrishna/FilesMath464/S23/Software/Lec19_03212023_Session.txt

Bland's rule (min-index rule)

1. Choose smallest j with $c_j' < 0$ as the pivot column (x_j enters).
2. If tied for leaving variable, choose the variable with the smallest index i to leave.

We will come back to Bland's rule later.

Finding Initial bfs

1. Two-phase simplex
2. big-M method (combines the two phases of Two-phase simplex).

$$\left.\begin{array}{l} \min\ \bar{c}^T\bar{x} \\ \text{s.t. } A\bar{x}=\bar{b} \\ \qquad \bar{x}\geqslant\bar{0} \end{array}\right\} \qquad \begin{array}{l} \min\ \bar{c}^T\bar{x}+M\sum_{i=1}^{m} y_i \\ \text{s.t. } A\bar{x}+I\bar{y}=\bar{b} \\ \qquad \bar{x}\geqslant\bar{0},\ \bar{y}\geqslant\bar{0} \end{array} \qquad \bar{y}\in\mathbb{R}^m$$

We assume $\bar{b}\geqslant\bar{0}$. If some $b_i<0$ to start with, scale that constraint i by -1, so that new $b_i>0$ (before adding $I\bar{y}$).

The y_i's are called **artificial variables**, one per constraint. They are artificial as their purpose is only to provide a starting bfs.

$M \rightarrow$ "big-M" acts like $+\infty$, can be considered a huge positive number. We can compare expressions involving M (unlike ones using ∞):

$$3M+2-(4M-10) = -M+12$$

$$-M+10{,}000 < 2M-8.$$

Since the min-objective function has My_i term for each i, it helps to set $y_i=0\ \forall i$, if possible. Thus, if we obtain an optimal solution with $y_i=0\ \forall i$, the corresponding x values are an optimal solution to the original LP. But if any $y_i>0$ in the optimal solution, the objective function value is essentially $+\infty$, indicating the original LP is infeasible.

Let's try the big-M method on our favorite LP:

$$\begin{array}{lllll} \min & 2x_1+x_2 & & & \\ \text{s.t.} & x_1+x_2-x_3 & & = 2 & x_6 \\ & 3x_1+x_2 & -x_4 & = 4 & x_7 \\ & 3x_1+2x_2 & +x_5 & = 10 & x_8 \\ & x_j \geq 0 \ \forall j & & & \end{array}$$

$\rightarrow$ instead of y_1, y_2, y_3, we use x_6, x_7, x_8.

$$\begin{array}{llllllll} \min & 2x_1+x_2 & & & +Mx_6+Mx_7+Mx_8 & & & \\ \text{s.t.} & x_1+x_2-x_3 & & & +x_6 & & & = 2 \\ & 3x_1+x_2 & -x_4 & & & +x_7 & & = 4 \\ & 3x_1+2x_2 & & +x_5 & & & +x_8 & = 10 \\ & & & x_j \geq 0 \ \forall j & & & & \end{array}$$

$$A = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \bar{b} = \begin{bmatrix} 2 \\ 4 \\ 10 \end{bmatrix}, \quad \bar{c}^T = [2 \ 1 \ 0 \ 0 \ 0 \ M \ M \ M].$$

We can choose $\{x_6, x_7, x_8\}$ as the starting bfs. $\bar{c}_B^T = [M \ M \ M]$.

$\rightarrow$ the artificial vars, in general.

$$B = I = B^{-1}. \quad \bar{x}_B = \begin{bmatrix} x \\ x_7 \\ x_8 \end{bmatrix} = B^{-1}\bar{b} = \bar{b} = \begin{bmatrix} 2 \\ 4 \\ 10 \end{bmatrix}.$$

$$\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A = [2 \ \ 1 \ \ 0 \ \ 0 \ \ 0 \ \ M \ \ M \ \ M] - [7M \ \ 4M \ \ -M \ \ -M \ \ M \ \ M \ \ M \ \ M]$$

$$= [-7M+2 \ \ -4M+1 \ \ M \ \ M \ \ -M \ \ 0 \ \ 0 \ \ 0]$$

$$-\bar{c}_B^T \bar{x}_B = -16M.$$

See the course web page for the Matlab session:

http://www.math.wsu.edu/faculty/bkrishna/FilesMath464/S23/Software/Lec19_03212023_Session.txt

MATH 464 – Lecture 20 (03/22/2018)

Today: * big-M method to detect infeasibility
* one full example
* revised simplex v/s tableau simplex

Detecting infeasibility

min $2x_1+x_2$
s.t. $x_1+x_2=4$ x_3
$2x_1+2x_2=10$ x_4
$x_1,x_2 \geq 0$

min $2x_1+x_2+Mx_3+Mx_4$
s.t. $x_1+x_2+x_3 = 4$
$2x_1+2x_2+x_4 = 10$
$x_j \geq 0 \ \forall j$

$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix}$ $\bar{b} = \begin{bmatrix} 4 \\ 10 \end{bmatrix}$, $\bar{c}^T = [2 \ 1 \ M \ M]$. Starting basis $\mathcal{B}=\{3,4\}$.
$\Rightarrow \bar{c}_B^T = [M \ M]$.

$B = B^{-1} = I$, $\bar{x}_B = \bar{b}$

$$\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A$$
$$= [2 \ 1 \ M \ M] - [M \ M]\begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix}$$
$$= [-3M+2 \ \ -3M+1 \ \ 0 \ \ 0].$$

$$z = \bar{c}_B^T \bar{x}_B = [M \ M]\begin{bmatrix} 4 \\ 10 \end{bmatrix} = 14M.$$

	$-14M$	$-3M+2$	$-3M+1$	0	0
$x_3=$	4	1	1	1	0
$x_4=$	10	2	2	0	1
	$-2M-4$	1	0	$3M-1$	0
$x_2=$	4	1	1	1	0
$x_4=$	2	0	0	-2	1

$R_0-(-3M+1)R_1$

→ optimal!

The optimal solution is $x_2=4$, $x_4=2$, with $z^*=2M+4$. Since an artificial variable (x_4) is >0 (basic), the original LP is infeasible.

Caution!: An artificial variable that is basic, but still $=0$ does not indicate infeasibility of original LP!

Two-phase Method

Given $\left\{\begin{array}{l} \min\ \bar{c}^T\bar{x} \\ \text{s.t. } A\bar{x}=\bar{b} \\ \qquad \bar{x}\geq\bar{0} \end{array}\right\}$ we write the following LP:

$\rightarrow \bar{b}\geq\bar{0}$

$$\left.\begin{array}{ll} \min & \bar{1}^T\bar{y}=\sum_{i=1}^{m} y_i \\ \text{s.t.} & A\bar{x}+I\bar{y}=\bar{b} \\ & \bar{x},\bar{y}\geq\bar{0} \end{array}\right\} \text{Phase 1 LP.}$$

↳ is guaranteed to be feasible!

$\bar{1}$: vector of 1's.

Here, y_i is the artificial variable for constraint i. Assuming $\bar{b}\geq\bar{0}$, which can be ensured before standardizing the LP by scaling any constraint with $b_i<0$ by -1, $\{\bar{y}=\bar{b},\ \bar{x}=\bar{0}\}$ is a feasible solution. Thus, the Phase 1-LP is feasible. Further, it is guaranteed to have an optimal solution: the minimum value of $\sum_{i=1}^{m} y_i$ is 0, when $y_i=0\ \forall i$, as $y_i\geq0\ \forall i$.

If the optimal objective function value of the phase-1 LP is zero, i.e., $y_i=0\ \forall i$ in the optimal solution, then the original LP is feasible. The corresponding $\bar{x}$ variables give a starting bfs for the phase 2 LP, which goes back to $\min\{\bar{c}^T\bar{x}\mid A\bar{x}=\bar{b},\ \bar{x}\geq0\}$.

On the other hand, if any $y_i>0$ in the optimal solution of the Phase 1 LP, the original LP is infeasible.

Phase-1 LP might be a better option than the big-M method if we're interested in only checking whether the original LP is feasible.

A full example

$$\max \bar{c}^T\bar{x} \equiv \min -\bar{c}^T\bar{x}$$

$$\begin{aligned}
\max \quad & z = -2x_1 + 3x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 \le 9 \\
& -(2x_1 + 5x_2 \ge -6) \\
& x_2 \ge 1 \\
& x_1 \text{ urs}, \; x_2 \ge 0
\end{aligned}$$

$x_1 \to (x_1 - x_3)$

so that rhs is $\geq 0 \Rightarrow -2x_1 - 5x_2 \le 6$

Steps:

1. Change objective to min, if needed.
2. Scale constraints with $b_i < 0$ by -1.
3. Take care of urs and ≤ 0 vars.
4. Convert to standard form.
5. Add artificial vars as needed.
6. Proceed with simplex method.

$$\begin{aligned}
\min \quad z = 2(x_1 - x_3) - 3x_2 \qquad\qquad\qquad\qquad & + Mx_7 \\
x_1 - x_3 + 3x_2 + x_4 \qquad\qquad\qquad & = 9 \\
-2x_1 + 2x_3 - 5x_2 \qquad + x_5 \qquad\qquad & = 6 \\
x_2 \qquad\qquad\qquad - x_6 + x_7 & = 1
\end{aligned}$$

We could add artificial variables for constraints 1 and 2 as well; but if a slack var is present, we might as well use them in the starting basis!

For large LP instances with tens of thousands of constraints, most of which are $\le$, it is wasteful to add artifical variables unless necessary — we could use the slack variables in the starting bfs!

$$\begin{array}{cccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \bar{c}^T = & [2 & -3 & -2 & 0 & 0 & 0 & M] \end{array}$$

$B = I = B^{-1}$ in starting tableau with $\mathcal{B} = \{4,5,7\}$, $B^{-1} = I$.

$$\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A \qquad \bar{c}_B = [0\ 0\ M], \quad \bar{x}_B = B\bar{b} = \bar{b} = \begin{bmatrix} 9 \\ 6 \\ 1 \end{bmatrix}.$$

$$-\bar{z} = -\bar{c}_B^T \bar{x}_B = -M.$$

We'll calculate this vector directly in Matlab!

See Matlab session at

http://www.math.wsu.edu/faculty/bkrishna/FilesMath464/S23/Software/Lec20_03232023_Session.txt

Optimal solution of the transformed LP: $x_2 = 24$, $x_3 = 63$, $x_6 = 23$, $z^* = -198$.

For the original LP, we had $x_1 = x_1 - x_3$, and had changed the max $\bar{c}^T\bar{x}$ to min $-\bar{c}^T\bar{x}$. Hence the optimal solution to the original LP is $x_1 = -63$, $x_2 = 24$ $(x_6 = 23)$, with $z^* = 198$.

Comparing Full Tableau and Revised Simplex Methods

Revised Simplex

1. Storage:

$$\underset{m \times m}{B^{-1}},\ \underset{m \times n}{A},\ \bar{c}'$$

B⁻¹ is not sparse, but A often is (especially in large real-life LPs). And $B^{-1}A$ is not sparse. In practice n is often much larger than m.

2. pivots / operations

Performed on B^{-1} $(m \times m)$, followed by $\bar{p}^T = \bar{c}_B^T B^{-1}$ and $\bar{c}'$.

Could compute c_j' one at a time.

Tableau Simplex

$$T = \left[\begin{array}{c|c} -z_B & \bar{c}'^T \\ \hline \bar{x}_B & B^{-1}A \end{array}\right],\ \bar{c}'$$

$(m+1) \times (n+1)$

Operations are performed on

$$T = \left[\begin{array}{c|c} -z & \bar{c}'^T \\ \hline x_B & B^{-1}A \end{array}\right]$$

$(m+1) \times (n+1)$

In summary, revised simplex is more efficient for large LPs.

The final project will ask you to compare your implementations of the revised simplex and the tableau simplex methods.

MATH 464 – Lecture 21 (03/27/2018)

Today: * LP duality
* primal-dual relationships
* ~~weak duality~~ didn't get to it

Duality

Consider $\left\{ \begin{array}{ll} \min & x^2+y^2 \\ \text{s.t.} & x+y=1 \end{array} \right\}$

We know how to optimize functions of the form $f(x,y)$ in calculus. With that idea in mind, we try to include the constraint $x+y=1$ in a modified function.

Unconstrained minimization in Calculus: $\min\limits_{x,y} f(x,y)$

$\left.\dfrac{\partial f}{\partial x}=0, \ \dfrac{\partial f}{\partial y}=0 \right\} \Rightarrow$ critical points

$\dfrac{\partial^2 f}{\partial x^2}>0 \quad \dfrac{\partial^2 f}{\partial y^2}>0$: local minimum

We write the **Lagrangian** $L(x,y,p) = x^2+y^2+p(1-x-y)$

price or penalty

p = price for not satisfying the constraint.

Minimize L: $\dfrac{\partial L}{\partial x}=0, \ \dfrac{\partial L}{\partial y}=0$ (p is a given constant)

$\Rightarrow \left.\begin{array}{l} 2x-p=0 \\ 2y-p=0 \end{array}\right\} \ x=y=\frac{p}{2}$. To satisfy $x+y=1$, we need $p=1$.

If we used $p(x+y-1)$ instead, we get $p=-1$.

$\Rightarrow x=y=\frac{1}{2}$ is the solution (critical point).

Indeed, $\dfrac{\partial^2 L}{\partial x^2}=\dfrac{\partial^2 L}{\partial y^2}=2$, and hence $x=y=\frac{1}{2}$ is a local minimum.

In fact, it is the global minimum here!

Consider $\left\{\begin{array}{l} \min \ \bar{c}^T\bar{x} \\ \text{s.t. } A\bar{x}=\bar{b} \\ \qquad \bar{x}\geqslant \bar{0} \end{array}\right\}$ (P) Assume $\bar{x}^*$ is an optimal solution (i.e., $A\bar{x}^*=\bar{b}$, $\bar{x}^*\geqslant\bar{0}$)

Convert (P) to a relaxed problem by choosing a vector of prices $\bar{p}$.
→ not unconstrained yet, as we need $\bar{x}\geqslant\bar{0}$

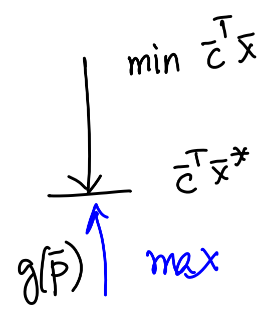
$$\min_{\bar{x}\geqslant\bar{0}} \left[\bar{c}^T\bar{x} + \bar{p}^T(\bar{b}-A\bar{x})\right] \text{ ———— } (P')$$

Let $g(\bar{p})$ be the optimal cost of (P') as a function of $\bar{p}$.
Notice $\bar{p}\in\mathbb{R}^m$. We get

$$\mathbf{g(\bar{p}) \leq \bar{c}^T\bar{x}^*} \text{ for any } \bar{p}.$$

$$g(\bar{p}) = \min_{\bar{x}\geqslant\bar{0}}\left[\bar{c}^T\bar{x}+\bar{p}^T(\bar{b}-A\bar{x})\right] \leqslant \bar{c}^T\bar{x}^* + \bar{p}^T\underbrace{(\bar{b}-\cancel{A}\bar{x}^*)}_{=\bar{0}, \text{ as } \bar{x}^* \text{ is feasible}}$$
$$= \bar{c}^T\bar{x}^*$$

So, $g(\bar{p})$ is a lower bound for $\bar{c}^T\bar{x}$ (in (P)) for any choice of $\bar{p}$. Our goal is to find the largest lower bound, i.e., $\max g(\bar{p})$.



In general, we want largest lower bounds and smallest upper bounds.

For many optimization problems, we are often satisfied with good bounds on the value of objective function. Of course, for LP problems, we have algorithms that are guaranteed to terminate – either with an optimal solution, or with the certificate of infeasibility or unboundedness. But for harder classes of optimization problems such as integer programs (IPs), we are often satisfied with finding bounds. For example we may find two solutions with values of objective function, say, 65 and 73, with the knowledge that the optimal objective function is in between the two values.

In this process of finding lower and upper bounds on the objective function, if we find lower bound = upper bound, we know that this is indeed the optimal value.

Coming back to the question of choosing an appropriate price vector $\bar{p}$, we want to find one such that $g(\bar{p})$ value is as large as possible – we are minimizing $\bar{c}^T\bar{x}$, and since $g(\bar{p})$ is a lower bound on $\bar{c}^T\bar{x}$ for any $\bar{p}$, we would like to push the lower bound as high as possible. In other words, we would like to maximize $g(\bar{p})$.

We will see later from the dual theorem that when $g(\bar{p}) = \bar{c}^T\bar{x}$, this value will be optimal for (P).

$$g(\bar{p}) = \min_{\bar{x} \geq \bar{0}} \left[\bar{c}^T \bar{x} + \bar{p}^T (\bar{b} - A\bar{x}) \right]$$

$$= \bar{p}^T \bar{b} + \min_{\bar{x} \geq \bar{0}} [\underbrace{\bar{c}^T \bar{x} - \bar{p}^T A \bar{x}}]$$

$$(\bar{c}^T - \bar{p}^T A)\bar{x} = \begin{cases} 0 & \text{if } \bar{c}^T - \bar{p}^T A \geq \bar{0} \\ -\infty & \text{if } \underbrace{\bar{c}^T - \bar{p}^T A \not\geq \bar{0}}_{\text{at least one entry} < 0} \end{cases}$$

We want to maximize $g(\bar{p})$, so we are interested only in cases where $\bar{c}^T - \bar{p}^T A \geq \bar{0}$, or $A^T \bar{p} \leq \bar{c}$. So we consider

$$\boxed{\begin{array}{l} \max \ \bar{p}^T \bar{b} \\ \text{s.t. } \bar{p}^T A \leq \bar{c}^T \end{array}}$$

(D) This is the **dual LP**.

could write also as $A^T \bar{p} \leq \bar{c}$.

(D) is the **dual linear program** to the original LP, which we refer to as the primal LP, (P). Here is the pair of LPs

$$\text{(P)} \quad \begin{array}{ll} \min & \bar{c}^T \bar{x} \\ \text{s.t.} & A\bar{x} = \bar{b} \\ & \bar{x} \geq 0 \end{array} \qquad\qquad \begin{array}{ll} \max & \bar{b}^T \bar{p} \\ \text{s.t.} & A^T \bar{p} \leq \bar{c} \end{array} \quad \text{(D)}$$

Given any LP we could write its corresponding dual LP in the same fashion. In particular, we need not necessarily start with an LP in standard form.

If, instead of $A\bar{x}=\bar{b}$ in (P), we had $A\bar{x}\geq\bar{b}$, we can transform it to

$$A\bar{x}-I\bar{s}=\bar{b} \Rightarrow [A\ -I]\begin{bmatrix}\bar{x}\\ \bar{s}\end{bmatrix}=\bar{b}.$$
$$\bar{s}\geq\bar{0}$$

We get $\bar{p}^T[A\ -I] \leq [\bar{c}^T\ \bar{0}^T]$ → m-vector of zeros $\equiv \bar{s}$

$$\Rightarrow \bar{p}^T A \leq \bar{c}^T$$
$$\bar{p}\geq\bar{0} \quad (-\bar{p}^T\leq\bar{0}^T)$$

So (P) $\left\{\begin{array}{ll}\min & \bar{c}^T\bar{x}\\ \text{s.t.} & A\bar{x}\geq\bar{b}\\ & \bar{x}\geq\bar{0}\end{array}\right\}$ has the dual $\left\{\begin{array}{ll}\max & \bar{p}^T\bar{b}\\ \text{s.t.} & \bar{p}^TA\leq\bar{c}^T\\ & \bar{p}\geq\bar{0}\end{array}\right\}$ (D).

We could specify the dual LP for a general form LP:

$\bar{a}_i^T$: i^{th} row of A $\qquad M_1\cup M_2\cup M_3=\{1,2,\ldots,m\}$.

$\bar{A}_j$: j^{th} column of A $\qquad N_1\cup N_2\cup N_3=\{1,2,\ldots,n\}$.

(P)
$$\begin{array}{lll}\min & \bar{c}^T\bar{x} & \\ \text{s.t.} & \bar{a}_i^T\bar{x}\geq b_i, & i\in M_1\\ & \bar{a}_i^T\bar{x}\leq b_i, & i\in M_2\\ & \bar{a}_i^T\bar{x}= b_i, & i\in M_3\\ & x_j\geq 0, & j\in N_1\\ & x_j\leq 0, & j\in N_2\\ & x_j \text{ urs}, & j\in N_3\end{array}$$

(D)
$$\begin{array}{lll}\max & \bar{p}^T\bar{b} & \\ \text{s.t.} & p_i\geq 0, & i\in M_1\\ & p_i\leq 0, & i\in M_2\\ & p_i \text{ urs}, & i\in M_3\\ & \bar{p}^T A_j\leq c_j, & j\in N_1\\ & \bar{p}^T A_j\geq c_j, & j\in N_2\\ & \bar{p}^T A_j= c_j, & j\in N_3\end{array}$$

Table of primal-dual relationships

We could specify all these correspondences in a table as follows.

Primal	min	max	Dual
variables	≥ 0	$\leq$	constraints
	≤ 0	$\geq$	
	urs	$=$	
constraints	$\geq$	≥ 0	variables
	$\leq$	≤ 0	
	$=$	urs	

normal constraints and vars

min LP – $\geq$ constraint is normal
min cost s.t. meet demand

max-LP – $\leq$ constraint is normal
max revenue s.t. limited resources

≥ 0 variables are always normal

The table of primal dual relationships could be presented in several equivalent forms – you need not memorize any one form! Here are the general rules.

→ variable in the primal LP $\Leftrightarrow$ constraint in dual LP.

→ normal (opposite to normal) variable in primal LP $\Leftrightarrow$ normal (opposite to normal) constraint in dual LP.

→ urs variable in primal LP $\Leftrightarrow$ $=$ constraint in dual LP.

normal vars : ≥ 0

normal constraints :
- $\leq$ for **max-LP** {max revenue s.t. upper bound on raw materials
- $\geq$ for **min-LP** {min cost s.t. make at least so many # products, i.e., meet demand.

Let's consider an example, and apply these relationships directly:

$$(P)\quad \begin{array}{lll} \max & 5x_1+4x_2-3x_3 & \\ \text{s.t.} & x_1 \qquad -5x_3 \geq 4 & y_1 \leq 0 \\ & 3x_1+x_2+2x_3 \leq 5 & y_2 \geq 0 \\ & x_1 \geq 0,\ x_2 \text{ urs},\ x_3 \geq 0 & \\ & \geq \quad = \quad \geq & \end{array}$$

$$\begin{array}{ll} \min & 4y_1+5y_2 \\ \text{s.t.} & y_1+3y_2 \geq 5 \\ & y_2 = 4 \\ & -5y_1+2y_2 \geq -3 \\ & y_1 \leq 0,\ y_2 \geq 0 \end{array}\quad (D)$$

If we take the dual of the dual LP, we get back the primal LP:

$$(D)\quad \begin{array}{lll} \min & 4y_1+5y_2 & \\ \text{s.t.} & y_1+3y_2 \geq 5 & u_1 \geq 0 \\ & y_2 = 4 & u_2 \text{ urs} \\ & -5y_1+2y_2 \geq -3 & u_3 \geq 0 \\ & y_1 \leq 0,\ y_2 \geq 0 & \\ & \geq \quad \leq & \end{array}$$

$$\begin{array}{ll} \max & 5u_1+4u_2-3u_3 \\ \text{s.t.} & u_1 \qquad -5u_3 \geq 4 \\ & 3u_1+u_2+2u_3 \leq 5 \\ & u_1 \geq 0,\ u_2 \text{ urs},\ u_3 \geq 0 \end{array}\quad (D')$$

$$(D')\quad \begin{array}{ll} \max & 5u_1+4u_2-3u_3 \\ \text{s.t.} & u_1 \qquad -5u_3 \geq 4 \\ & 3u_1+u_2+2u_3 \leq 5 \\ & u_1 \geq 0,\ u_2 \text{ urs},\ u_3 \geq 0 \end{array}$$

$$\begin{array}{ll} \max & 5x_1+4x_2-3x_3 \\ \text{s.t.} & x_1 \qquad -5x_3 \geq 4 \\ & 3x_1+x_2+2x_3 \leq 5 \\ & x_1 \geq 0,\ x_2 \text{ urs},\ x_3 \geq 0 \end{array}\quad (P)$$

Notice (D') is equivalent to (P)!

MATH 464 – Lecture 22 (03/29/2018)

Today:
* duality in matrix form
* weak duality
* strong duality

Table of primal-dual relationships

Primal	max	min	Dual
constraints	$\leq$	≥ 0	variables
	$\geq$	≤ 0	
	$=$	urs	
variables	≥ 0	$\geq$	constraints
	≤ 0	$\leq$	
	urs	$=$	

Duality in Matrix Form

(P)
$$\begin{aligned} \min\ & \bar{c}^T \bar{x} \\ \text{s.t. } & A\bar{x} \leq \bar{b} \quad \bar{p} \leq \bar{0} \\ & \quad = \end{aligned}$$

(D)
$$\begin{aligned} \max\ & \bar{b}^T \bar{p} \\ \text{s.t. } & A^T \bar{p} = \bar{c} \\ & \bar{p} \leq \bar{0} \end{aligned}$$

(P)
$$\begin{aligned} \max\ & \bar{c}^T \bar{x} + \bar{d}^T \bar{y} \\ \text{s.t. } & A\bar{x} + B\bar{y} = \bar{f} \quad \bar{p}\text{ urs} \\ & \bar{x} \geq \bar{0},\ \bar{y} \leq \bar{0} \\ & \quad \geq \quad\ \leq \end{aligned}$$

(D)
$$\begin{aligned} \min\ & \bar{f}^T \bar{p} \\ \text{s.t. } & A^T \bar{p} \geq \bar{c} \\ & B^T \bar{p} \leq \bar{d} \\ & \bar{p}\text{ urs} \end{aligned}$$

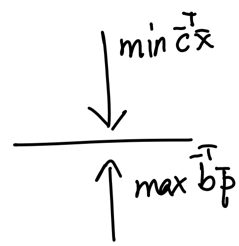
Weak Duality (BT-ILO Theorem 4.3)

Consider

$$(P)\quad \begin{array}{ll} \min & \bar{c}^T\bar{x} \\ \text{s.t.} & A\bar{x} = \bar{b} \quad \bar{p} \\ & \bar{x} \geqslant \bar{0} \\ & \quad \leqslant \end{array} \qquad \text{and} \qquad \begin{array}{ll} \max & \bar{b}^T\bar{p} \\ \text{s.t.} & A^T\bar{p} \leq \bar{c} \\ & \bar{p} \text{ urs} \end{array} \quad (D).$$

Let $\bar{x}$ be feasible for (P) and $\bar{p}$ be feasible for (D). Then $\mathbf{\bar{b}^T\bar{p} \leq \bar{c}^T\bar{x}}$.

Recall the intuitive picture of pulling $\bar{c}^T\bar{x}$ down from above, and pushing $\bar{b}^T\bar{p}$ up from below.



The values of $\bar{c}^T\bar{x}$ for any feasible $\bar{x}$ always lie above the values of $\bar{b}^T\bar{p}$ for any feasible $\bar{p}$.

Proof $\quad \bar{p}$ is feasible for (D) gives

$$\left(A^T\bar{p} \leq \bar{c}\right)^T \qquad \text{take transpose on both sides}$$

$$\Longrightarrow \quad \bar{p}^TA \leq \bar{c}^T$$

$\bar{x}$ is feasible for (P) gives $\bar{x} \geqslant \bar{0}$, $A\bar{x} = \bar{b}$.

$$\left(\bar{p}^TA \leq \bar{c}^T\right)\bar{x} \qquad \text{multiply by } \bar{x} \text{ on the right (on both sides)}$$

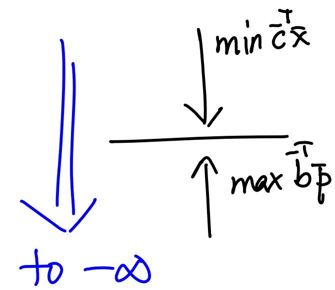
$$\Longrightarrow \quad \bar{p}^T\underbrace{A\bar{x}}_{\bar{b}} \leq \bar{c}^T\bar{x} \qquad \text{stays} \leq \text{as } \bar{x} \geqslant \bar{0}$$

$$\Longrightarrow \quad \bar{p}^T\bar{b} \leq \bar{c}^T\bar{x}. \qquad \square$$

We get the following two corollaries.

Corollary 4.1 If the primal optimal cost is $-\infty$, then the dual is infeasible. Similarly, if the dual cost is $+\infty$, then the primal is infeasible.

Note that the primal (dual) is unbounded when its optimal cost is $-\infty$ $(+\infty)$.



More generally, if an LP is unbounded, its dual LP is infeasible. Going back to the intuitive picture, if we can keep pulling down $\bar{c}^T\bar{x}$ without limit, then there could be no finite lower bound given by any $\bar{b}^T\bar{p}$, i.e., there is no feasible $\bar{p}$ for dual (D).

Corollary 4.2 If $\bar{x}, \bar{p}$ are feasible for (P) and (D), respectively, and $\bar{b}^T\bar{p} = \bar{c}^T\bar{x}$, then $\bar{x}$ and $\bar{p}$ are optimal for (P) and (D), respectively.

Proof $\bar{b}^T\bar{p} = \bar{c}^T\bar{x}$. Weak duality gives that $\bar{b}^T\bar{p} \leq \bar{c}^T\bar{y}$ for any $\bar{y}$ that is a feasible solution for (P). Hence we get $\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y}$ for any $\bar{y}$ that is a feasible solution for (P), i.e., $\bar{x}$ is optimal. A similar argument holds for (D).

Strong Duality (BT-ILO Theorem 4.4)

If an LP has an optimal solution, then so does its dual, and the optimal costs are equal.

Proof Let $\bar{x}$ be optimal for (P) $\{\min \bar{c}^T\bar{x} \mid A\bar{x}=\bar{b}, \bar{x} \geq \bar{0}\}$.
Let B be the corresponding optimal basis matrix. Then $\bar{x}_B = B^{-1}\bar{b}$.

The reduced costs for (P) must be non-negative, from the optimality conditions for (P).

$$\Rightarrow \quad \bar{c}'^T = \bar{c}^T - \underbrace{\bar{c}_B^T B^{-1}}_{\bar{p}^T} A \geq \bar{0}^T$$

Define $\bar{p}^T = \bar{c}_B^T B^{-1} \Longrightarrow \left(\bar{c}^T - \bar{p}^T A \geq \bar{0}^T\right)^T$

$\Rightarrow \quad A^T\bar{p} \leq \bar{c}$, i.e., $\bar{p}$ is feasible for (D).

Also, $\bar{b}^T\bar{p} = \bar{p}^T\bar{b} = \bar{c}_B^T B^{-1}\bar{b} = \bar{c}_B^T\bar{x}_B = \bar{c}^T\bar{x}$.
Hence, by weak duality (Corollary 4.2), $\bar{p}$ is optimal for (D). $\square$

Possibilities for (P) and (D)

Dual

Primal	finite optimum	unbounded	infeasible
finite optimum	✓	✗	✗
unbounded	✗	✗	✓
infeasible	✗	✓	✓

An 'X' means that combination is not possible. For instance, we cannot have an infeasible primal LP for which the dual LP has a finite optimum.

Both (P) and (D) could be infeasible:

$$(P)\quad \begin{aligned} \min\ & x_1 + 2x_2 \\ \text{s.t.}\ & x_1 + x_2 = 2 \quad p_1 \\ & 3x_1 + 3x_2 = 4 \quad p_2 \\ & \quad = \qquad = \end{aligned}$$

$$\begin{aligned} \max\ & 2p_1 + 4p_2 \\ \text{s.t.}\ & p_1 + 3p_2 = 1 \\ & p_1 + 3p_2 = 2 \end{aligned} \quad (D)$$

MATH 464 — Lecture 23 (04/04/2023)

Today: * complementary slackness conditions
* interpretation of duality
* economic interpretation of dual

Duality:

$$(P)\quad \begin{array}{ll} \min & \bar{c}^T\bar{x} \\ \text{s.t.} & A\bar{x}=\bar{b} \quad \bar{p} \\ & \bar{x}\geqslant \bar{0} \end{array} \qquad\qquad \begin{array}{ll} \max & \bar{b}^T\bar{p} \\ \text{s.t.} & A^T\bar{p}\leq \bar{c} \end{array} \quad (D)$$

Complementary Slackness Conditions (CSCs) (BT-ILO Theorem 4.5)

min LP in general form

Let $\bar{x},\bar{p}$ be feasible for (P) and (D), respectively. They are optimal iff

$$p_i(\bar{a}_i^T\bar{x}-b_i)=0 \quad \forall\, i \ (i=1,\dots,m)$$
$$(c_j-\bar{p}^TA_j)\,x_j=0 \quad \forall\, j \ (j=1,\dots,n) \quad \text{hold.}$$

Proof Let $u_i=p_i(\bar{a}_i^T\bar{x}-b_i)$ and $v_j=(c_j-\bar{p}^TA_j)\,x_j$.

By the definition of the dual LP, we get $u_i\geqslant 0\ \forall i$, as explained below.

If constraint i is $\bar{a}_i^T\bar{x}\geqslant b_i$ then $p_i\geqslant 0 \implies u_i\geqslant 0$, and
if constraint i is $\bar{a}_i^T\bar{x}\leq b_i$ then $p_i\leq 0 \implies u_i\geqslant 0$.

But if constraint i is $\bar{a}_i^T\bar{x}=b_i$, then $u_i=0$ (p_i is urs, but it does not matter).

Hence $u_i\geqslant 0\ \forall i$ holds. Similarly, $v_j\geqslant 0\ \forall j$.

So, $$\sum_{i=1}^m u_i=\sum_{i=1}^m p_i(\bar{a}_i^T\bar{x}-b_i)=\bar{p}^T(A\bar{x}-\bar{b})=\bar{p}^TA\bar{x}-\bar{p}^T\bar{b}, \quad\text{and}$$
$$\sum_{j=1}^n v_j=\bar{c}^T\bar{x}-\bar{p}^TA\bar{x}.$$

$$\implies \sum_{i=1}^m u_i+\sum_{j=1}^n v_j=\bar{c}^T\bar{x}-\bar{p}^T\bar{b}=0 \text{ iff } \bar{x},\bar{p} \text{ are optimal for}$$
(P) and (D), by strong duality.

Hence we have $u_i \geq 0 \; \forall i$; $v_j \geq 0 \; \forall j$, $\sum_{i=1}^{m} u_i + \sum_{j=1}^{n} v_j = 0$ at optimality, which means $u_i = 0 \; \forall i$ and $v_j = 0 \; \forall j$ at optimality (iff). $\square$

If a constraint is **not** active, its corresponding dual variable is zero at optimality. Similarly, if a variable is **nonzero** at optimality, its corresponding dual constraint is active (at optimality).

Caution: The reverse implication might not hold, i.e., if a constraint is active, its dual variable could be zero or nonzero.

<u>An Example</u>: Consider the following pair of primal-dual LPs:

$$
(P) \quad \begin{aligned}
\max \quad & 3x_1 + 4x_2 + x_3 + 5x_4 \\
\text{s.t.} \quad & x_1 + 2x_2 + x_3 + 2x_4 \leq 5 \quad p_1 \geq 0 \\
& 2x_1 + 3x_2 + x_3 + 3x_4 \leq 8 \quad p_2 \geq 0 \\
& x_j \geq 0 \;\; \forall j \\
& \geq
\end{aligned}
$$

$$
(D) \quad \begin{aligned}
\min \quad & 5p_1 + 8p_2 \\
\text{s.t.} \quad & p_1 + 2p_2 \geq 3 \quad \text{—(1)} \\
& 2p_1 + 3p_2 \geq 4 \quad \text{—(2)} \\
& p_1 + p_2 \geq 1 \quad \text{—(3)} \\
& 2p_1 + 3p_2 \geq 5 \quad \text{—(4)} \\
& p_1, p_2 \geq 0
\end{aligned}
$$

It is given that constraints (1) and (4) are active in an optimal solution of (D). Using that fact, and CSCs, solve (P).

(1) and (4) are active $\Rightarrow \left.\begin{aligned} p_1 + 2p_2 &= 3 \\ 2p_1 + 3p_2 &= 5 \end{aligned}\right\} \; p_1 = p_2 = 1.$

At $p_1 = 1, p_2 = 1$, Constraints (2) and (3) are not active, $x_2 = x_3 = 0$ in (P). by CSCs

Also, $p_1 > 0, p_2 > 0 \Rightarrow$ constraints in (P) are active.

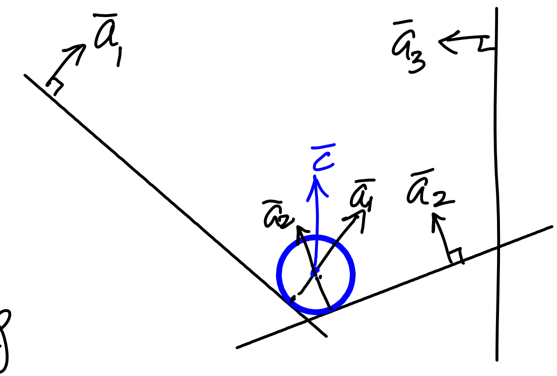
$\Rightarrow$ In (P) at optimality, $\left.\begin{aligned} x_1 + 2x_4 &= 5 \\ 2x_1 + 3x_4 &= 8 \end{aligned}\right\} \Rightarrow x_1 = 1, x_4 = 2$ $(x_2 = x_3 = 0)$

Indeed $\bar{c}^T \bar{x} = \bar{p}^T \bar{b} = 13$, and hence we have solved (P) and (D).

An intuitive, mechanical analogy of duality

Consider the primal-dual pair of LPs:

$$(P)\quad \begin{array}{ll} \min & \bar{c}^T\bar{x} \\ \text{s.t.} & A\bar{x} \geqslant \bar{b} \end{array} \qquad\qquad \begin{array}{ll} \max & \bar{p}^T\bar{b} \\ \text{s.t.} & \bar{p}^T A = \bar{c}^T \\ & \bar{p} \geqslant \bar{0} \end{array} \quad (D)$$



Consider a bowl determined by the constraints $\bar{a}_i^T\bar{x} \geqslant b_i$, and a ball that is being pulled down by gravity ($\equiv \min\ \bar{c}^T\bar{x}$). The ball comes to rest at the lowest point in the bowl.

At equilibrium, the force provided by the walls touching the ball balance gravity (i.e., its weight). The forces excerted by the walls act perpendicular to the walls. Let p_i be the force applied by wall i. Hence we have $\sum_{i=1}^{m} p_i\bar{a}_i = \bar{c}$ for some non-negative magnitudes of forces p_i.

These equilibrium p_i's are an optimal solution to (D). Also, $p_i = 0$ if the wall i (represented by $\bar{a}_i$) does not touch the ball at equilibrium. So, if $\bar{a}_i^T\bar{x} > b_i$, $p_i = 0$; which is the CSC for that constraint-dual variable pair. Hence we get that

$$p_i(\bar{a}_i^T\bar{x} - b_i) = 0 \quad \forall i.$$

$$\Rightarrow \bar{p}^T\bar{b} = \sum_i p_i b_i = \sum_i p_i \bar{a}_i^T\bar{x} = \underbrace{\bar{p}^T A}_{\bar{c}^T,\ \text{from (D)}}\bar{x} = \bar{c}^T\bar{x}, \text{ confirming}$$

that $\bar{x}$ and $\bar{p}$ are optimal for (P) and (D), respectively.

23.4

Economic Interpretation

Exercise 1.15 A company produces two kinds of products. A product of the first type requires 1/4 hours of assembly labor, 1/8 hours of testing, and \$1.2 worth of raw materials. A product of the second type requires 1/3 hours of assembly, 1/3 hours of testing, and \$0.9 worth of raw materials. Given the current personnel of the company, there can be at most 90 hours of assembly labor and 80 hours of testing, each day. Products of the first and second type have a market value of \$9 and \$8, respectively.

(a) Formulate a linear programming problem that can be used to maximize the daily profit of the company.

Here is the LP:

3. Let x_j = # units of product j made, for $j = 1, 2$. Here is the LP.

$$
\begin{array}{llcll}
\max & (9-1.2)x_1 + (8-0.9)x_2 & & & \text{(daily profit)} \\
\text{s.t.} & (1/4)\,x_1 + (1/3)\,x_2 & \leq & 90 & \text{(assembly hours)} \quad p_1 \geq 0 \\
& (1/8)\,x_1 + (1/3)\,x_2 & \leq & 80 & \text{(testing hours)} \quad p_2 \geq 0 \\
& x_1, x_2 & \geq & 0 & \text{(non-negativity)} \\
& \geq \ \geq & & &
\end{array}
$$

And here is the dual LP:

$$
\begin{array}{lll}
\min & 90p_1 + 80p_2 & (\ ?\ ) \\
\text{s.t.} & (\tfrac{1}{4})p_1 + (\tfrac{1}{8})p_2 \geq 7.8 & (\ ?\ ) \\
& (\tfrac{1}{3})p_1 + (\tfrac{1}{3})p_2 \geq 7.1 & (\ ?\ ) \\
& p_1, p_2 \geq 0 & \text{(non-neg)}
\end{array}
$$

How can we interpret the objective function and constraints? We consider a scenario where a competing firm wants to buy out (the operations of) this company.

Let a competing firm consider buying out the company. It will have to pay for the hours of assembly and testing. Let p_1 and p_2 be the price per hour of assembly and testing, respectively, that this firm is offering. Since they are prices, they must be nonnegative.

The firm's offer must be attractive to the company. If the company has $\frac{1}{4}$ hr of assembly and $\frac{1}{8}$ hr of testing available, it can make one unit of product 1, which gives a profit of \$7.8. Hence, p_1 and p_2 should be such that $\left(\frac{1}{4}\right)p_1 + \left(\frac{1}{8}\right)p_2 \geq 7.8$, which is the first constraint. The second constraint is interpreted in a similar fashion for product 2.

The firm would like to minimize the total cost of this purchase, i.e., minimize $90p_1 + 80p_2$, which is the objective function.

We could interpret the (optimal) dual variables as "shadow prices" — more on this idea in the next lecture.

MATH464 – Lecture 24 (04/06/2023)

Today: * More on AMPL
* Dual variables as shadow prices
* Farkas' Lemma

Hw9 is now due on Monday, April 10.

AMPL session: **read** command to read in parameters from text files.
Display optimal dual variables using names of primal constraints.
See AMPL session for details:

https://www.math.wsu.edu/faculty/bkrishna/FilesMath464/S23/Software/AMPL/Lec1_01102023_Session.txt

---

Back to Economic Interpretation of Dual LP

The optimal solution values for p_1 and p_2 could be interpreted as **shadow prices** for assembly and testing hours.

The optimal solution for (D) is $p_1 = 31.2$, $p_2 = 0$.

At optimality in (P), we are using all of the 90 assembly hours available, but only 45 out of the 80 testing hours available. The company could consider buying more assembly hours. The company would be willing to pay up to $p_1 = \$31.2$ for each extra assembly hour.

But the amount the company would pay for an extra testing hour is $p_2 = 0$, as it is currently not using all testing hours available.

The second result is also a direct interpretation of complementary slackness. Since (testing hours) constraint is not active, its dual variable is zero at optimality. Since we are not using all of the 80 hrs of testing available, we will not pay anything for more testing hours.

Farkas' Lemma

Consider $\left\{ \begin{array}{l} A\bar{x}=\bar{b} \\ \bar{x}\geq\bar{0} \end{array} \right\}$. Does this system have a solution?

To answer this question completely, we need to justify our YES/NO response. If the answer is YES, we could specify an $\bar{x}$ that satisfies the system. Indeed, we could easily check whether $\bar{x}$ satisfies $A\bar{x}=\bar{b}$ and $\bar{x}\geq\bar{0}$.

What if our response is NO? How do we justify this response? Suppose there exists a $\bar{p}$ such that $\bar{p}^T A\geq\bar{0}, \bar{p}^T\bar{b}<0$. Then since $\bar{x}\geq\bar{0}$, for any potential solution $\bar{x}$, we get $\bar{p}^T \underbrace{A\bar{x}}_{\bar{b}} \geq 0$, which conflicts with $\bar{p}^T\bar{b}<0$. Hence, such a $\bar{p}$ could be provided as a certificate of infeasibility for the original system. Farkas' lemma provides such a result – either the original system is feasible, or an alternative system is feasible that certifies the infeasibility of the original system.

Farkas' lemma is a classic example of such "systems of alternatives" type result, which are seen in many subfields of mathematics. We could prove Farkas' lemma from first principles, but the proof would be harder than if we use LP duality.

(BT-ILO Theorem 4.6) Let $A \in \mathbb{R}^{m\times n}$, $\bar{b} \in \mathbb{R}^m$. Then exactly one of the following two statements hold.

(a) There exists some $\bar{x} \geq \bar{0}$ such that $A\bar{x} = \bar{b}$.

(b) There exists some $\bar{p}$ such that $\bar{p}^T A \geq \bar{0}$ and $\bar{p}^T \bar{b} < 0$.

Proof ($\Rightarrow$) If (a) holds, there exists $\bar{x} \geq \bar{0}$ such that $A\bar{x} = \bar{b}$.

$$\Rightarrow \bar{p}^T A \bar{x} = \bar{p}^T \bar{b}.$$

If $\bar{p}^T A \geq \bar{0}$ then $\bar{p}^T \bar{b} = \bar{p}^T (A\bar{x}) = (\bar{p}^T A)\bar{x} \geq 0$, as $\bar{x} \geq \bar{0}$.

$\Rightarrow$ (b) cannot hold.

If (b) holds, there exists $\bar{p}$ such that $\bar{p}^T A \geq \bar{0}$ and $\bar{p}^T \bar{b} < 0$. For any $\bar{x} \geq \bar{0}$, $(\bar{p}^T A)\bar{x} \geq 0$, but $\bar{p}^T \bar{b} < 0$ means $A\bar{x} \neq \bar{b}$.

So, (a) cannot hold.

This is the "easy" direction of the proof. For showing the reverse direction ($\Leftarrow$), we will use LP duality.

($\Leftarrow$) If (a) does not hold, we want to show (b) holds.

Let $\nexists\ \bar{x} \geq 0$ s.t. $A\bar{x} = \bar{b}$ ((a) does not hold).

Consider the following primal-dual pair of LPs:

$$(P)\quad \begin{array}{ll} \max & \bar{0}^T \bar{x} \\ \text{s.t.} & A\bar{x} = \bar{b} \quad \bar{p} \\ & \bar{x} \geq \bar{0} \\ & \geq \end{array} \qquad\qquad \begin{array}{ll} \min & \bar{p}^T \bar{b} \\ \text{s.t.} & \bar{p}^T A \geq \bar{0}^T \\ & \bar{p} \text{ urs} \end{array} \quad (D)$$

(P) is infeasible here. Hence (D) is infeasible or unbounded.
But $\bar{p}=\bar{0}$ is feasible for (D), and hence it is unbounded, i.e., $\bar{p}^T b \to -\infty$.
Hence some $\bar{p}$ exists that is feasible for (D), and $\bar{p}^T \bar{b} < 0$. Hence, (b) holds.

If (b) does not hold, then $\bar{p}^T \bar{b} \geq 0 \ \forall \ \bar{p}$ such that $\bar{p}^T A \geq \bar{0}$.
This condition means (D) is feasible and is not unbounded, i.e., it must have an optimal solution. This result follows from the observation that $\bar{p}^T \bar{b}$ cannot be decreased without limit.
By duality, (P) must have an optimal solution as well.
Hence (P) has a feasible solution, i.e., (a) holds. $\square$

Note how we came up with the primal-dual pair of LPs.
Since the rhs of the system in (b) is $\bar{0}$ (from $\bar{p}^T A \geq \bar{0}$), that is precisely the objective function of the primal LP (P).
Also, since the constraints in (P) are equations ($A\bar{x}=\bar{b}$), the dual variables $\bar{p}$ are urs.

MATH 464 – Lecture 25 (04/11/2023)

Today:
* illustration of Farkas' lemma
* asset pricing (application of Farkas' lemma).
* optimal dual prices as marginal costs.

On AMPL: For the inventory problem, best to set param n as the # months so that you can use 1..n or 0..n-1 or 0..n as the required sets to write, e.g., the balance constraint in one line:

s[i-1] + x[i] = Demand[i] + s[i]; for i = 1,..,n.

You could work with a set Months := m1 m2 ... m12; but then you can't directly index previous/next month: AMPL will **not** take m2+1 as m3!

Farkas' lemma

(a) $\exists\ \bar{x} \geqslant \bar{0}$ s.t. $A\bar{x} = \bar{b}$

(b) $\exists\ \bar{p}$ s.t. $\bar{p}^T A \geqslant \bar{0}$ and $\bar{p}^T\bar{b} < 0$.

Illustration in 2D

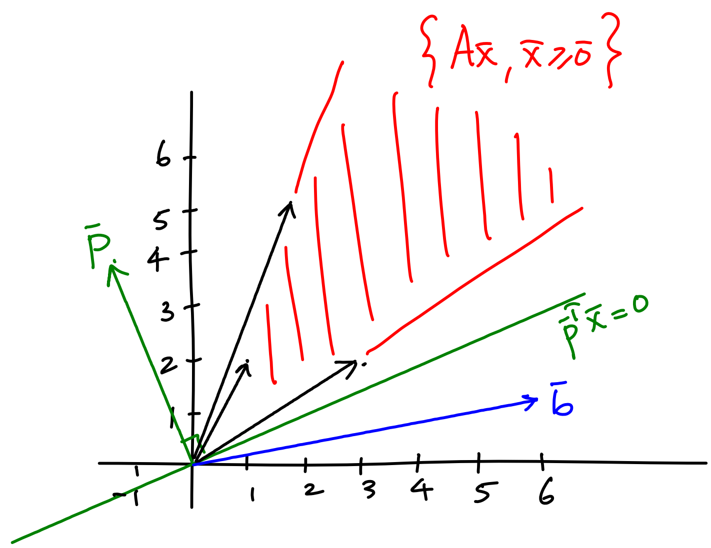
$$A = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 2 & 5 \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} 6 \\ 1 \end{bmatrix}.$$

Does there exist $\bar{x} \geqslant \bar{0}$ s.t. $A\bar{x} = \bar{b}$?

complicating part.

No, as seen in the picture!

$\bar{b}$ is not in the cone $\{A\bar{x} \mid \bar{x} \geqslant \bar{0}\}$.



For $\bar{p} = \begin{bmatrix} -1 \\ 4 \end{bmatrix}$, we get $\bar{p}^T A = [-1 \ \ 4]\begin{bmatrix} 1 & 3 & 2 \\ 2 & 2 & 5 \end{bmatrix} = [7 \ \ 5 \ \ 18] \geqslant \bar{0}$

and $\bar{p}^T\bar{b} = [-1 \ \ 4]\begin{bmatrix} 6 \\ 1 \end{bmatrix} = -2.$

We see that $\{A\bar{x} \mid \bar{x} \geqslant \bar{0}\}$ and $\bar{b}$ lie on opposite sides of $\{\bar{x} \mid \bar{p}^T\bar{x} = 0\}$. Hence the vector $\bar{p}$ gives a certificate of infeasibility of the original system. Note that there could be multiple such $\bar{p}$ vectors possible here – any solution to the alternative system in (b) will do.

In this example, note that the $\bar{x} \geq \bar{0}$ restriction is indeed the "complicating" constraint. If we wanted to check if $A\bar{x} = \bar{b}$ (for any $\bar{x}$), then answer would have been YES, since the columns of A span $\mathbb{R}^2$ here. But we could present Farkas' lemma in an equivalent form without $\bar{x} \geq \bar{0}$ in alternative (a).

Alternative forms of Farkas' lemma

Exactly one of the following two alternatives hold:

(a) $\exists \bar{x}$ s.t. $A\bar{x} = \bar{b}$.

(b) $\exists \bar{p}$ s.t. $\bar{p}^T A = \bar{0}$, $\bar{p}^T \bar{b} = -1$. $\longrightarrow$ similar to $[0\ 0 \dots 0 | \blacksquare]$ where $\blacksquare \neq 0$, which indicates $A\bar{x} = \bar{b}$ is inconsistent.

Idea: If (b) holds, we can derive (by taking linear combinations of rows) $0 = -1$ from $A\bar{x} = \bar{b}$, and hence (a) cannot hold.

Proof (part of).

($\Leftarrow$) If (a) does not hold, (b) holds.

$$(P) \quad \begin{array}{ll} \max & \bar{0}^T \bar{x} \\ \text{s.t.} & A\bar{x} = \bar{b} \quad \bar{p} \end{array} \qquad\qquad \begin{array}{ll} \min & \bar{p}^T \bar{b} \\ \text{s.t.} & \bar{p}^T A = \bar{0}^T \end{array} \quad (D)$$

(a) does not hold means (P) is infeasible. So (D) is either infeasible or unbounded. But $\bar{p} = \bar{0}$ is feasible for (D), so (D) is unbounded. Hence $\exists \bar{p}$ s.t. $\bar{p}^T A = \bar{0}^T$ and $\bar{p}^T \bar{b} < 0$. We can scale $\bar{p}$ such that $\bar{p}^T \bar{b} = -1$ holds. Hence (b) holds.

Application of Farkas' Lemma – Asset Pricing

There are n assets, and after a period of investment there are m possible states (or scenarios). Let r_{ij} be the return from asset j under state/scenario i. Hence if you invest \$1 in asset j, and the state turns out to be i, you expect to receive \$$r_{ij}$.

Assumption 1 r_{ij} is same for both $x_j \geq 0$ and $x_j < 0$.

($x_j < 0$ means you're in a "short" position — you sell $|x_j|$ at the start, with the promise to buy back at the end at rate r_{ij} if state is i. The payoff of $r_{ij}x_j < 0$ here, meaning you have to pay $r_{ij}|x_j|$).

Assumption 1 might not hold in general, but that's not too critical for our illustration here.

Let $R = [r_{ij}] \in \mathbb{R}^{m \times n}$, and $\bar{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ be the amount of assets at start ($x_j \geq 0$ and $x_j < 0$ are both possible).

Let $\bar{w}$ represent the wealth at the end of the period. Then we write $\bar{w} = R\bar{x}$. Let $\bar{p}$ be the prices at the start for each asset. Then $\bar{p}^T\bar{x}$ is the total cost of the portfolio.

An **arbitrage** is a price such that $R\bar{x} \geq \bar{0}$ but $\bar{p}^T\bar{x} < 0$, i.e., we get a non-negative return for a negative price.

Assumption 2 Prices will equilibriate to avoid arbitrages.

"No arbitrages" means "if $R\bar{x} \geq \bar{0}$ then $\bar{p}^T\bar{x} \geq \bar{0}$".

(BT-ILO Theorem 4.8) There exist no arbitrages iff there exists $\bar{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \geq \bar{0}$ such that $\bar{p} = R^T\bar{q}$, $\bar{q} \geq \bar{0}$.

Intuitively, this theorem says that there will be no arbitrages iff the prices are some non-negative linear combination of the return rates $R = [r_{ij}]$. This result is usually presented independent of any connections to linear programming. But we can prove it by a direct application of Farkas' lemma.

Proof Follows from Farkas' lemma

(a) $\exists \bar{x} \geq \bar{0}$ s.t. $A\bar{x} = \bar{b}$.
(b) $\exists \bar{p}$ s.t. $\bar{p}^T A \geq \bar{0}$, $\bar{p}^T\bar{b} < 0$.

We construct one alternative to represent arbitrages. The other alternative then presents the criterion given in the theorem for existence of $\bar{q}$.

(a) $\bar{R}^T\bar{q} = \bar{p}$, $\bar{q} \geq \bar{0}$

(b) $R\bar{x} \geq \bar{0}$, $\bar{p}^T\bar{x} < 0$ (arbitrage)

Optimal dual variables as marginal costs

$$(P)\quad \begin{aligned} \min\ & \bar{c}^T \bar{x} \\ \text{s.t. } & A\bar{x} = \bar{b} \\ & \bar{x} \geq \bar{0} \end{aligned}$$

Assume (P) has optimal solution $\bar{x}^*$ that is non-degenerate, and let B be the corresponding basis matrix.

Then $\bar{x}_B = B^{-1}\bar{b} > \bar{0}$. Optimal cost is $\bar{c}_B^T \bar{x}_B$.

Also, the reduced costs: $\bar{c}'^T = \bar{c}^T - \bar{c}_B^T B^{-1} A \geq \bar{0}$ (optimality conditions).

Consider changing $\bar{b}$ to $\bar{b}+\bar{d}$ (rhs vector). Since $\bar{x}_B > \bar{0}$, for $\bar{d}$ small enough, the new optimal solution $\bar{x}_B = B^{-1}(\bar{b}+\bar{d}) \geq \bar{0}$. And hence the new optimal cost is $\bar{c}_B^T B^{-1}(\bar{b}+\bar{d}) = \underbrace{\bar{c}_B^T B^{-1}\bar{b}}_{\text{original cost}} + \underbrace{\bar{c}_B^T B^{-1}}_{\bar{p}^T}\bar{d}$.

Hence we get new optimal cost = old optimal cost + $\bar{p}^T\bar{d}$. Thus $\bar{p}$ can be interpreted as the marginal cost (or shadow price) for increasing each rhs entry by 1 unit. In particular, p_i is the shadow price for "resource" i, i.e., the price to pay for increasing b_i to b_i+1.

Dual Simplex Method

Caution! This is **NOT** applying simplex method to the dual LP!

Tableau for the primal simplex method:

$$(P)\quad \begin{array}{ll} \min & \bar{c}^T\bar{x} \\ \text{s.t.} & A\bar{x}=\bar{b} \\ & \bar{x}\geq\bar{0} \end{array} \qquad\qquad \begin{array}{ll} \max & \bar{p}^T\bar{b} \\ \text{s.t.} & \bar{p}^T A\leq\bar{c}^T \end{array}\quad (D)$$

$\bar{c}' \geq 0$ (maintain)

$-\bar{c}_B^T\bar{x}_B$	$\bar{c}^T-\bar{c}_B^T B^{-1}A$
$B^{-1}\bar{b}$	$B^{-1}A$

≥ 0

allow <0
strive to make them ≥ 0.

more in the next lecture...

MATH 464 – Lecture 26 (04/13/2023)

Today: * dual simplex method
* proof exercises from HW7

Dual Simplex Method

Tableau for the primal simplex method:

$-\bar{c}_B^T \bar{x}_B$	$\bar{c}^T - \bar{c}_B^T B^{-1} A$
$B^{-1}\bar{b}$	$B^{-1}A$

$\bar{c}'^T$

(P) $\min\ \bar{c}^T \bar{x}$ s.t. $A\bar{x} = \bar{b}$, $\bar{x} \geq \bar{0}$

$\max\ \bar{p}^T \bar{b}$ s.t. $\bar{p}^T A \leq \bar{c}^T$ (D)

Optimality Conditions $\quad \bar{x}_B = B^{-1}\bar{b} \geq \bar{0}$ (feasibility) and

(optimality) $\bar{c}'^T = \bar{c}^T - \underbrace{\bar{c}_B^T B^{-1}}_{\bar{p}^T} A \geq \bar{0}^T \Rightarrow \bar{p}^T A \leq \bar{c}^T$

So, optimality for (P) $\Longleftrightarrow$ feasibility for (D)

Primal cost $= \bar{c}_B^T \bar{x}_B = \underbrace{\bar{c}_B^T B^{-1}}_{\bar{p}^T} \bar{b} = \bar{p}^T \bar{b} =$ dual cost

If $\bar{c}'^T \geq \bar{0}^T$, we have dual feasibility. And since the costs are equal, the solutions $\bar{x}$ and $\bar{p}$ are optimal for (P) and (D), respectively.

In primal simplex, we maintain primal feasibility $(B^{-1}\bar{b} \geq \bar{0})$, and we strive for primal optimality $(\bar{c}'^T \geq \bar{0}^T)$. In dual simplex, we maintain dual feasibility, i.e., $\bar{c}'^T \geq \bar{0}^T$, and strive for dual optimality $(B^{-1}\bar{b} \geq \bar{0})$.

dual simplex:

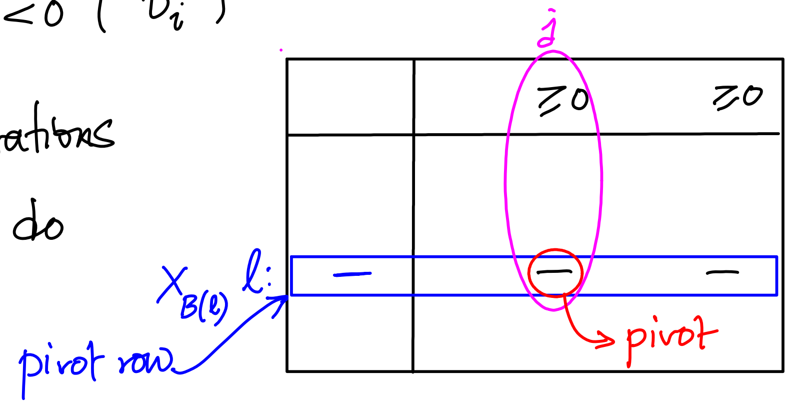
$-\bar{c}_B^T \bar{x}_B$	$\bar{c}^T - \bar{c}_B^T B^{-1} A$
$B^{-1}\bar{b}$	$B^{-1}A$

$\bar{c}' \geq \bar{0}$ (maintain)

could be < 0

So, entries in Column-0 (Rows 1 to m) could be < 0 in dual simplex. If they are < 0, we "pivot them out".

Let $x_{B(\ell)} < 0$, and let the ℓ^{th} row be $(x_{B(\ell)}, v_1, \ldots, v_n)$. We take this ℓ^{th} row as the **pivot row**. For all $v_i < 0$, we find $\frac{c_i'}{|v_i|} = \frac{-c_i'}{v_i}$. Let $j = \underset{v_i<0}{\operatorname{argmin}} \left\{ \frac{-c_i'}{v_i} \right\}$. Then x_j enters, and $x_{B(\ell)}$ leaves. The pivoting operations then are similar to ones we do in primal simplex.



We convert column j to $\begin{bmatrix} 0 \\ \hline \bar{e}_\ell \end{bmatrix}$, where $\bar{e}_\ell = \ell^{th}$ unit vector.

Note that when we scale the pivot row to make the pivot entry equal to 1, the $x_{B(\ell)}$ value will necessarily become > 0, as the pivot entry $a_{\ell j}$ is necessarily < 0 to start with, and so is $x_{B(\ell)}$.

<u>Example to illustrate a dual-simplex pivot</u>

$-6/-2 \quad -10/-3 \longrightarrow$ min-ratio computations

		x_1	x_2	x_3	x_4	x_5
	0	2	6	10	0	0
$x_4 =$	2	−2	4	1	1	0
$x_5 =$	−1	4	−2	−3	0	1
	−3	14	0	1	0	3
$x_4 =$	0	6	0	−5	1	2
$x_2 =$	1/2	−2	1	3/2	0	−1/2

$R_2 \times (-1/2)$

We now have both primal and dual optimality!

Here is how the various pieces connect:

$$\begin{array}{ccc} \text{primal feasibility} & \xrightarrow{\text{primal simplex}} & \text{primal optimality} \\ \equiv & & \equiv \\ \text{dual optimality} & \xleftarrow{\text{dual simplex}} & \text{dual feasibility} \end{array}$$

We take the dual path, maintaining dual feasibility and striving for dual optimality.

Consider the following LP:

$$\begin{array}{llll} \min & 5x_1 + 35x_2 + 20x_3 & & \text{slack variables} \\ \text{s.t.} & x_1 - x_2 - x_3 & \leq -2 & x_4 \\ & -x_1 - 3x_2 & \leq -3 & x_5 \\ & x_j \geq 0 & & \end{array}$$

If we were to use primal simplex, we would add 2 excess and 2 artificial variables. Instead, we could start with the obvious basis using the two slack variables and do dual simplex!

		x_1	x_2	x_3	x_4	x_5
	0	5	35	20	0	0
$x_4 =$	-2	1	-1	-1	1	0
$x_5 =$	-3	-1	-3	0	0	1
	-15	0	20	20	0	5
$x_4 =$	-5	0	-4	-1	1	1
$x_1 =$	3	1	3	0	0	-1
	-40	0	0	15	5	10
$x_2 =$	$5/4$	0	1	$1/4$	$-1/4$	$-1/4$
$x_1 =$	$-3/4$	1	0	$-3/4$	$3/4$	$-1/4$
	-55	20	0	0	20	5
$x_2 =$	1	$1/3$	1	0	0	$-1/3$
$x_3 =$	1	$-4/3$	0	1	-1	$1/3$

Standard solvers such as CPLEX uses some heuristics to identify which variant (primal or dual) of simplex method to use. We often see the dual simplex being used.

Proof-type problems from Hw7

Exercise 3.2 (Optimality conditions) Consider the problem of minimizing $\mathbf{c}'\mathbf{x}$ over a polyhedron P. Prove the following:

(a) A feasible solution $\mathbf{x}$ is optimal if and only if $\mathbf{c}'\mathbf{d} \geq 0$ for every feasible direction $\mathbf{d}$ at $\mathbf{x}$.

not necessarily a bfs

($\Rightarrow$) Start with optimal $\bar{x}$, show $\bar{c}^T\bar{d} \geq 0$ $\forall$ feasible directions $\bar{d}$.

($\Leftarrow$) Start with $\bar{c}^T\bar{d} \geq 0$ $\forall$ feasible directions $\bar{d}$ at $\bar{x}$, show $\bar{x}$ is optimal.

($\Rightarrow$) Let $\bar{x}$ be an optimal solution. Hence $\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y}$ $\forall\, \bar{y} \in P$.
Let $\bar{d}$ be a feasible direction at $\bar{x}$. Then there is some $\theta > 0$ such that $\underbrace{\bar{x}+\theta\bar{d}}_{\bar{y}} \in P$ (i.e., $\bar{x}+\theta\bar{d}$ is feasible).

Since $\bar{x}$ is optimal, $\cancel{\bar{c}^T\bar{x}} \leq \bar{c}^T(\cancel{\bar{x}}+\theta\bar{d}) \Rightarrow \bar{c}^T(\theta\bar{d}) \geq 0$,
but since $\theta > 0$, we get $\bar{c}^T\bar{d} \geq 0$.

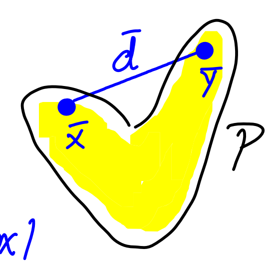
($\Leftarrow$) Let $\bar{c}^T\bar{d} \geq \bar{0}$ $\forall$ feasible directions $\bar{d}$ at $\bar{x}$. $\rightarrow$ $\bar{x}+\theta\bar{d} \in P$ for some $\theta > 0$.

Let $\bar{y} \in P$. We want to show $\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y}$ (i.e., $\bar{x}$ is optimal).

We write $\bar{y} = \bar{x} + (\bar{y}-\bar{x}) = \bar{x} + \theta(\bar{y}-\bar{x})$ where $\theta = 1$.

We want to argue $\bar{y}-\bar{x}\,(=\bar{d})$ is a feasible direction at $\bar{x}$. Indeed $\bar{y}-\bar{x}$ is a feasible direction at $\bar{x}$, as both $\bar{x}, \bar{y} \in P$ and P is convex. Writing $\bar{d} = \bar{y}-\bar{x}$, we get $\bar{c}^T\bar{d} \geq \bar{0} \Rightarrow$
$\bar{c}^T\bar{x} \leq \bar{c}^T\bar{y}$. This applies to any $\bar{y} \in P$, so $\bar{x}$ is optimal.

$\bar{d} = (\bar{y}-\bar{x})$ may not be feasible if P is not convex!



Exercise 3.3 Let $\mathbf{x}$ be an element of the standard form polyhedron $P = \{\mathbf{x} \in \Re^n \mid \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$. Prove that a vector $\mathbf{d} \in \Re^n$ is a feasible direction at $\mathbf{x}$ if and only if $\mathbf{Ad} = \mathbf{0}$ and $d_i \geq 0$ for every i such that $x_i = 0$.

($\Rightarrow$) Let $\bar{d}$ be a feasible direction at $\bar{x}$. By definition, $\exists\, \theta > 0$ such that $\bar{x} + \theta \bar{d} \in P$. $\bar{x} \in P$ to start with, so $A\bar{x} = \bar{b}$. Also, we have $A(\bar{x} + \theta\bar{d}) = \bar{b}$. Hence $\theta A\bar{d} = \bar{0}$, which along with $\theta > 0$ gives $A\bar{d} = \bar{0}$. Further, $\bar{x} + \theta\bar{d} \geq \bar{0}$, which gives $\theta d_i \geq 0$ when $x_i = 0$, and since $\theta > 0$, we get $d_i \geq 0$ $\forall i$ s.t. $x_i = 0$.

($\Leftarrow$) Let $\bar{d}$ be such that $A\bar{d} = \bar{0}$ and $d_i \geq 0$ $\forall i$ with $x_i = 0$.

We have $A(\bar{x} + \theta\bar{d}) = A\bar{x} = \bar{b}$ for any θ, as $\bar{x} \in P$.

Further, when $x_i = 0$, $x_i + \theta d_i \geq 0$ for all $\theta \geq 0$. When $x_i > 0$, $x_i + \theta d_i \geq 0$ for small enough $\theta > 0$ (think min-ratio test).

Hence for some $\theta > 0$, $\bar{x} + \theta\bar{d} \in P$, i.e., $\bar{d}$ is a feasible direction at $\bar{x}$.

Since $x_i > 0$, assume $d_i < 0$ (else the result is trivial). We are scaling d_i by $\theta > 0$, though, and hence $x_i + \theta d_i \geq 0$ for θ small enough. Now, we consider all $x_i > 0$, and take the smallest $\theta > 0$ for all i.

This result gives a characterization of all feasible directions $\bar{d}$ at $\bar{x}$: every such direction satisfies $A\bar{d} = \bar{0}$ and $d_i \geq 0$ when $x_i = 0$.

MATH464 — Lecture 27 (04/18/2023)

Today: * Bland's rule in Matlab
* Hw and other problems from BT-ILO

Bland's rule

Simply put, this is the "minimum index rule" — the non-basic x_j with $c_j' < 0$ and smallest j enters, and in case of a tie, the basic variable x_ℓ with the smallest ℓ (that ties) leaves. Since we order the variables $x_1, \dots, x_n$ by default, choosing the entering variable is easy — just pick the leftmost one. But choosing the leaving variable, while easy to do by hand, will require a bit more work.

We will maintain and update the basis $\mathcal{B}$ (stored as Bind in Matlab). See the session from today's lecture for details:

https://www.math.wsu.edu/faculty/bkrishna/FilesMath464/S23/Software/Lec27_04182023_Session.txt

Now let's consider a problem not assigned in the homework.

Exercise 3.19 While solving a standard form problem, we arrive at the following tableau, with x_3, x_4, and x_5 being the basic variables:

$$
\begin{array}{c|c}
 & \begin{array}{cccccc} & & x_2 & & & \end{array} \\
 & \begin{array}{|c|ccccc|}
\hline
-10 & \delta & -2^{=0} & 0 & 0 & 0 \\
\hline
4 & -1 & \eta & 1 & 0 & 0 \\
1 & \alpha & -4 & 0 & 1 & 0 \\
\beta{=}0 & \gamma & 3 & 0 & 0 & 1 \\
\hline
\end{array}
\end{array}
\quad \xrightarrow{\frac{2}{3}} \quad R_0 + \left(\tfrac{2}{3}\right) R_3
$$

x_4

$$\delta + \left(\tfrac{2}{3}\right)\gamma$$

The entries α, β, γ, δ, η in the tableau are unknown parameters. For each one of the following statements, find some parameter values that will make the statement true.

(a) The current solution is optimal and there are multiple optimal solutions.

(b) The optimal cost is $-\infty$.

(c) The current solution is feasible but not optimal.

(a) Notice x_2 could enter (as $c_2' = -2 < 0$). Thus the current bfs is optimal only if x_2 could enter but not change the cost. In other words, we need a degenerate bfs here. If $\beta = 0$ and $\gamma > 0$, we could get $\theta^* = 0$ (min-ratio). Hence x_2 could enter without changing the cost. The EROs give $c_1' = \delta + \left(\frac{2}{3}\right)\gamma$. If $\delta + \left(\frac{2}{3}\right)\gamma \geq 0$, the resulting tableau is optimal, and hence so is the current solution, thus giving multiple optimal solutions. $\boxed{\beta = 0, \gamma > 0, \delta + \frac{2}{3}\gamma \geq 0.}$

(b) x_2 cannot improve the cost without bound, as we need $\beta \geq 0$ for feasibility. We get unbounded LP with $\delta < 0$, $\alpha \leq 0, \gamma \leq 0$ ($\beta \geq 0$ is needed for feasibility).

(c) We need $\beta > 0$, as then we get $\theta^* = \min\left(\frac{\beta}{3}, \frac{4}{\eta}\right)$ if $\eta > 0$. $\theta^* > 0$ here, and hence x_2 could enter to improve the solution. We also need either $\delta \geq 0$, or if $\delta < 0$, then $\alpha > 0$ or $\gamma > 0$. Another option is $\delta < 0$, and $\alpha < 0, \gamma < 0$, when the LP will be unbounded.

Hw10 Problems

Exercise 4.3 The purpose of this exercise is to show that solving linear programming problems is no harder than solving systems of linear inequalities.

Suppose that we are given a subroutine which, given a system of linear inequality constraints, either produces a solution or decides that no solution exists. Construct a simple algorithm that uses a single call to this subroutine and which finds an optimal solution to any linear programming problem that has an optimal solution.

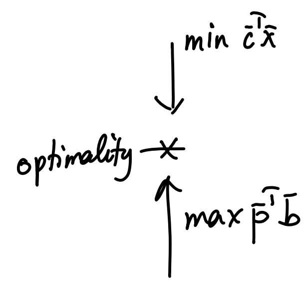
Subroutine: feasibility of a system of linear inequalities.

Use LP duality to come up with a system of linear inequalities that you could input to the subroutine only once.

$$\text{(P)}\quad \begin{aligned} \min\ & \bar{c}^T\bar{x} \\ \text{s.t. } & A\bar{x} = \bar{b} \\ & \bar{x} \geq \bar{0} \end{aligned} \qquad\qquad \begin{aligned} \max\ & \bar{p}^T\bar{b} \\ \text{s.t. } & \bar{p}^T A \leq \bar{c} \end{aligned}\quad \text{(D)}$$

Hint: weak/strong duality.

$\hookrightarrow$ $\bar{c}^T\bar{x} \geq \bar{p}^T\bar{b}$ for feasible $\bar{x}, \bar{p}$ for (P) and (D).



If $\bar{c}^T\bar{x} = \bar{p}^T\bar{b}$, then $\bar{x}, \bar{p}$ are optimal for (P) and (D), respectively.

Consider the system

$$\boxed{\begin{aligned} A\bar{x} &= \bar{b} \\ \bar{x} &\geq \bar{0} \\ \bar{p}^T A &\leq \bar{c} \\ \bar{c}^T\bar{x} &= \bar{p}^T\bar{b} \end{aligned}} \quad \begin{aligned} &\Big\}\ \text{(P) feasibility} \\ &\rightarrow \text{(D) feasibility} \\ &\rightarrow \text{optimality} \end{aligned}$$

Call subroutine once with this system as input.

Bonus: Think about how to handle the situation if the subroutine says the system is infeasible!

Farkas' lemma:

Exercise 4.26 Let $\mathbf{A}$ be a given matrix. Show that exactly one of the following alternatives must hold.

(a) There exists some $\mathbf{x} \neq \mathbf{0}$ such that $\mathbf{Ax} = \mathbf{0}$, $\mathbf{x} \geq \mathbf{0}$.

(b) There exists some $\mathbf{p}$ such that $\mathbf{p}'\mathbf{A} > \mathbf{0}'$. we want an equivalent system that is $\geq$, not $>$.

The crux is to come up with an appropiate pair of primal-dual LPs, and follow arguments presented in class.

$$\text{(P)} \quad \begin{array}{l} \max \bar{1}^T \bar{x} \\ A\bar{x} = \bar{0} \\ \bar{x} \geq \bar{0} \end{array} \qquad\qquad \begin{array}{l} \min \bar{p}^T \bar{0} \\ \bar{p}^T A \geq \bar{1} \end{array} \quad \text{(D)}$$

Suppose (b) holds. Then (D) is feasible. (D) is not unbounded, as $\bar{p}^T\bar{0} = 0$ for any $\bar{p}$. So (D) has optimal cost $= 0$. Hence for (P), we get that the only optimal solution is $\bar{x} = \bar{0}$. (as $\bar{1}^T\bar{x} = \sum_i x_i = 0$ optimal $\Rightarrow \bar{x} = \bar{0}$ is the only optimal solution).

So (a) cannot hold.

MATH464 — Lecture 28 (04/20/2023)

Today: Project

Simplex Implementations

Q. What about starting bfs?

Add an artificial variable per row, and add $+M$ to the cost vector.
Choose all artificial varibles in your starting bfs.

Default format of functions desired:

$[\text{status}, z^*, \bar{x}^*, \text{nItr}] = \text{TableauSimplex}(A, \bar{b}, \bar{c}, \text{EntRule}, \text{LvgRule})$

status: optimal/unbounded/infeasible

nItr: # iterations

EntRule: option for entering variable

LvgRule: option for leaving variable

These options could be set as integers, e.g., 1, 2, ⋯, and procedures chosen within your program based on them.

Running Time

Could use tic+toc or use clock() as shown below.

```
t1=tic;
RevisedSimplex;
RunTime = toc(t1);
```

OR

```
t1 = clock();
TableauSimplex;
t2 = clock();
Elapsed_Time = etime(t2,t1);
```

28.2

3. (5) Jimbo Enterprises produces n products. Each product can be produced in one of m machines. Let t_{ij} be the time in hours needed to produce one unit of product i on machine j. For month k, the number of hours available on machine j is h_{kj}. Customers are willing to buy **up to** d_{ik} units of product i in month k at the unit cost of c_{ik}. Formulate an LP that Jimbo can use to maximize the revenue by selling the products for the next p months.

Let x_{ijk} = # units of product i made on machine j in month k (and sold)

$$\max \quad z = \sum_{i=1}^{n} \sum_{k=1}^{p} c_{ik} \left(\sum_{j=1}^{m} x_{ijk} \right) \qquad \text{(total revenue)}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} t_{ij} x_{ijk} \le h_{kj}, \quad j=1,\dots,m, \quad k=1,\dots,p \quad \text{(max hrs)}$$

$$\sum_{j=1}^{m} x_{ijk} \le d_{ik}, \quad i=1,\dots,n, \quad k=1,\dots,p \quad \text{(max prods)}.$$

$$x_{ijk} \ge 0 \quad \forall i,j,k \quad \text{(non-neg)}$$

We need to convert this LP to $A\bar{x} \le \bar{b}$ form (first).

What do $A, \bar{b}, \bar{c}$ look like?

Q: Before that... when will this LP be infeasible?

If h_{kj}, d_{ik}, t_{ij} are all ≥ 0, the LP is guaranteed to be feasible, as $\bar{x} = \bar{0}$ is feasible.

1. You could get an infeasible instance if some of h_{kj}, d_{ik}, t_{ij} are chosen as < 0. OR
2. Change the LP so that d_{ik}'s are demands and you're minimizing total cost (c_{ik}'s can be taken as unit costs). Then, if some d_{ik} are too large for chosen h_{kj} values, the LP could be infeasible.

Back to formulation

Let $n=2, m=2, p=3$.

$$\bar{x}^T = [x_{111}, x_{112}, x_{113}, x_{121}, x_{122}, x_{123}, x_{211}, x_{212}, x_{213}, x_{221}, x_{222}, x_{223}]$$

$$\bar{c}^T = [\underbrace{c_{11}, c_{12}, c_{13}, c_{11}, c_{12}, c_{13}}_{m \text{ times}}, c_{21}, c_{22}, c_{23}, c_{21}, c_{22}, c_{23}]$$

Data C is an $\underline{n \times p}$ matrix. Here $C \in \mathbb{R}^{2\times 3}$: $C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}$

Check out: **repmat** and **reshape** in Matlab.

$$C = \begin{bmatrix} 3 & 4 & 1 \\ 2 & 5 & 6 \end{bmatrix}$$

$$\bar{c}^T = [3\ 4\ 1\ 3\ 4\ 1\ 2\ 5\ 6\ 2\ 5\ 6];$$

$$\bar{x}^T = [x_{111}, x_{112}, x_{113}, x_{121}, x_{122}, x_{123}, x_{211}, x_{212}, x_{213}, x_{221}, x_{222}, x_{223}]$$

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \end{array}
\underbrace{\begin{bmatrix}
t_{11} & & & & & & t_{21} & & & & & \\
& t_{11} & & & & & & t_{21} & & & & \\
& & t_{11} & & & & & & t_{21} & & & \\
& & & t_{12} & & & & & & t_{22} & & \\
& & & & t_{12} & & & & & & t_{22} & \\
& & & & & t_{12} & & & & & & t_{22} \\
1 & & & 1 & & & & & & & & \\
& 1 & & & 1 & & & & & & & \\
& & 1 & & & 1 & & & & & & \\
& & & & & & 1 & & & 1 & & \\
& & & & & & & 1 & & & 1 & \\
& & & & & & & & 1 & & & 1
\end{bmatrix}}_{A}
\bar{x} \le
\bar{b} = \begin{bmatrix} h_{11} \\ h_{21} \\ h_{31} \\ h_{12} \\ h_{22} \\ h_{32} \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix}$$

Generating Data

Decide ranges of values for n, m, p. Start small, and increase in steps. First ensure your simplex functions work correctly.

For each given n, m, p, generate data (C, D, T, H) randomly.

You would want to go to large enough values so that you (start to) observe your revised simplex starting to gain on your tableau simplex.

MATH 464 – Lecture 29 (04/25/2023)

Today: * on the project
* Interior point methods
* Instances "bad" for simplex

More on the Project

* On input (data) format:

  ✗ Should not prompt user to input data (entry by entry).

  ✓ Should work just like linprog( ).

* You should have a way to either regenerate the same random instances — to solve again, or to solve using different approaches.
  Can either set the seed for random number generators in Matlab deterministically (i.e., record the seed)
  Or you could save the instances themselves.

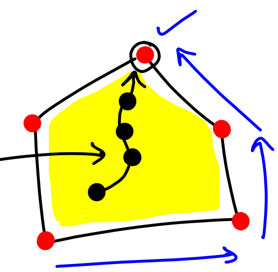
* Starting bfs, slack variables, and more...
  * your simplex functions are supposed to solve standard form LPs: $\min\{\bar{c}^T\bar{x} \mid A\bar{x}=\bar{b},\ \bar{x}\geqslant\bar{0}\}$.
  * If slack variables need to be added to get all constraints to equations, do so before/in the process of creating your A matrix.
  * May be simplest to add artificial variables with big-M to get starting bfs — works for all instances!

Do the project in MATLAB, especially if you have to ask me about doing it in another package/language (meaning you're not confident enough)!

Interior Point Methods

Simplex method → starts at a bfs, moves to adjacent bfs so that cost improves; repeat till termination.

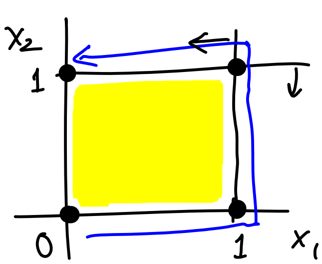
Moves along the boundary of the polyhedron.



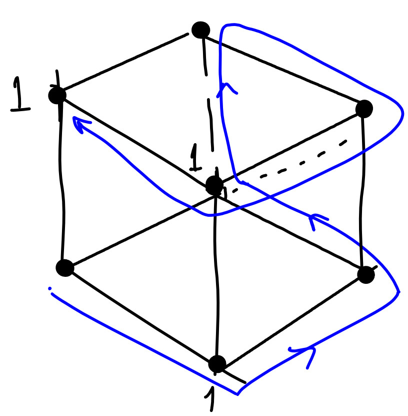
Interior point method → starts at a feasible solution (not necessarily a bfs), move inside the feasible region to improve the cost, till you terminate.

Instances that are "bad" for simplex method

A cube in $\mathbb{R}^n$: $0 \le x_j \le 1 \ \forall j$ $\quad 2^n$ vertices.



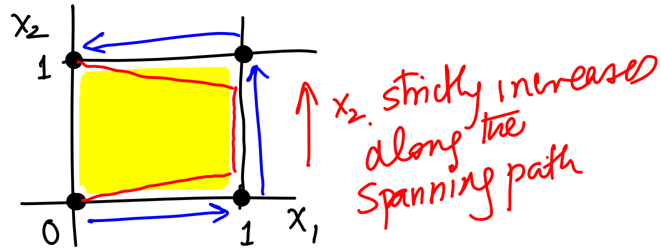
Spanning path: A path traveling along edges, visits every vertex.



Path visiting all 8 vertices of the cube.

If each vertex in this path improves the cost, then the simplex method visits 2^n vertices before it finishes!

We modify the cube a little bit to get this setting:



Let $\epsilon \in (0, 0.5)$. Consider the following set of constraints.

$$\epsilon \le x_1 \le 1$$

$$\epsilon x_{i-1} \le x_i \le 1 - \epsilon x_{i-1}, \quad i = 2, \dots, n.$$

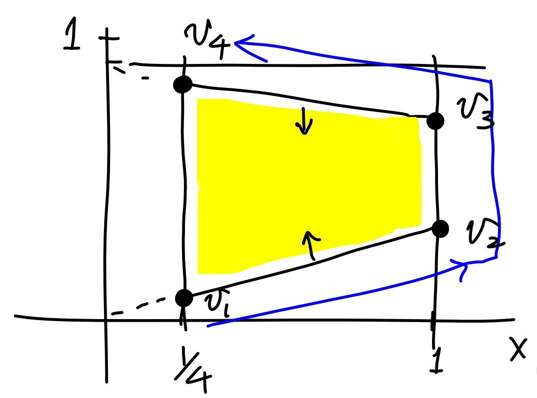
Here is the instance for $\epsilon = \frac{1}{4}$, $n = 2$.

$$\frac{1}{4} \le x_1 \le 1$$

$$\frac{1}{4} x_1 \le x_2 \le 1 - \frac{1}{4} x_1$$

$$x_1 - 4x_2 \le 0$$

$$x_1 + 4x_2 \le 4$$



x_2 increases strictly as we go $v_1 \to v_2 \to v_3 \to v_4$

In the general instance, if we were to maximize x_n, then the simplex method could indeed take 2^n iterations to terminate. Also, note that we get quite close to the unit cube by choosing $\epsilon > 0$ very small.

Interior point methods could take a much shorter path through the interior of the polytope.