IM2010: Operations Research
Preparation for the Simplex Method
(Chapter 4)

Ling-Chieh Kung

Department of Information Management
National Taiwan University

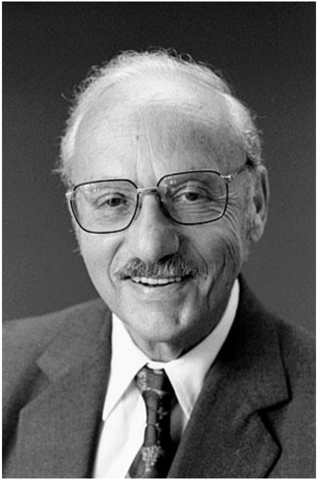
March 14, 2013

Introduction

- In this chapter, we will study **how to solve** a linear program.
- In fact, we will learn how to solve **any** linear program.
- The algorithm we will introduce is **the simplex method**.
  - Developed by **George Dantzig** in 1947.
  - Opened the whole field of Operations Research.
  - **Very efficient** for almost all practical linear programs.
  - With **very simple ideas**.
- It is not just a method to solve linear programs.
  - It discovers many important **properties** of linear programming.
  - It provides **insights** in solving other problems.
  - It shows the **beauty** of mathematics.

George Dantzig

- 1914 – 2005.
- A UC Berkeley Ph.D. (1946).
- A Stanford professor.
- Developed the simplex method when solving Air Force planning problems.
  - Each plan is called a program in US Air Force.



George Dantzig's doctoral dissertation

- Adopted from "Linear Programming: 1: Introduction" by Dantzig and Thapa.
  - "I owe a great debt to **Jerzy Neyman**, the leading mathematical statistician of his day, who guided my graduate work at Berkeley."
  - "My thesis was on two famous unsolved problems in mathematical statistics that I **mistakenly** thought were a homework assignment and solved."

George Dantzig's presentation

- Adopted from "Linear Programming: 1: Introduction" by Dantzig and Thapa.
  - In 1948, Dantzig summarized his works about Linear Programming in a conference. He explained how to formulate and solve linear programs.
  - After his presentation, Hotelling said: "But we all know the world is nonlinear."
  - Dantzig, a young unknown at that time, did not know how to response.
  - Von Neumann said: "The speaker titled his talk 'linear programming' and carefully stated his axioms. If you have an application that satisfies the axioms, well use it. If it does not, then don't."

Road map

- **Standard form linear programs**.
- Basic solutions.
- Basic feasible solutions.
- The idea of the simplex method.

Standard form linear programs

- As we know, linear programs may be of all kinds.
  - Maximization or minimization objective functions.
  - Equality, no-greater-than, and no-less-than constraints.
  - Nonnegative, nonpositive, and free variables.
- We will first show that all linear programs has an equivalent **standard form** representation.
- Then we will show how to use the simplex method to solve standard form linear programs.

Standard form linear programs

- First, let's define the standard form.

Definition 1 (Standard form linear program)

A linear program is in the standard form if

- *all the constraints RHS are nonnegative,*
- *all the variables are nonnegative, and*
- *all the constraints are equalities.*

- RHS = right hand sides. For any constraint

$$g(x) \leq b, \quad g(x) \geq b, \quad \text{or } g(x) = b,$$

  b is the RHS.
- There is no restriction on the objective function.

Standard form linear programs

- Why the following two LPs are not in the standard form?

$$\begin{array}{rrcrcr} \min & 3x_1 & + & 2x_2 & & \\ \text{s.t.} & x_1 & - & x_2 & \geq & 6 \\ & 2x_1 & + & x_2 & \leq & -4 \\ & \multicolumn{5}{l}{x_1 \geq 0, \quad x_2 \geq 0} \end{array} \qquad \begin{array}{rrcrcr} \max & 3x_1 & + & 2x_2 & & \\ \text{s.t.} & x_1 & - & x_2 & = & 6 \\ & 2x_1 & + & x_2 & = & 4 \\ & \multicolumn{5}{l}{x_1 \geq 0, \quad x_2 \leq 0} \end{array}$$

Finding the standard form

- How to find the standard form for a linear program?
- Requirement 1: **Nonnegative RHS**.
  - If it is negative, **switch** the LHS and the RHS.
  - E.g.,

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

is equivalent to

$$-2x_1 - 3x_2 \geq 4.$$

Finding the standard form

- Requirement 2: **Nonnegative variables**.
  - If x_i is **nonpositivie**, replace it by $-x_i$. E.g.,

$$2x_1 + 3x_2 \leq 4, x_1 \leq 0 \quad \Leftrightarrow \quad -2x_1 + 3x_2 \leq 4, x_1 \geq 0.$$

  - If x_i is **free**, replace it by $x_i' - x_i''$, where $x_i', x_i'' \geq 0$. E.g.,

$$2x_1 + 3x_2 \leq 4, x_1 \text{ urs.} \quad \Leftrightarrow \quad 2x_1' - 2x_1'' + 3x_2 \leq 4, x_1' \geq 0, x_1'' \geq 0.$$

$x_i = x_i' - x_i''$	$x_i' \geq 0$	$x_i'' \geq 0$
5	5	0
0	0	0
-8	0	8

Finding the standard form

- Requirement 3: **Equality constraints**.
  - For a less-than-or-equal-to constraint, **add a slack** variable. E.g.,

$$2x_1 + 3x_2 \leq 4 \quad \Leftrightarrow \quad 2x_1 + 3x_2 + x_3 = 4, \quad x_3 \geq 0.$$

  - For a greater-than-or-equal-to constraint, **minus a surplus/excess** variable. E.g.,

$$2x_1 + 3x_2 \geq 4 \quad \Leftrightarrow \quad 2x_1 + 3x_2 - x_3 = 4, \quad x_3 \geq 0.$$

  - For ease of exposition, they will both be called slack variables.
  - A slack variable measures the **gap** between the LHS and the RHS of a constraint.
  - Why nonnegative?

An example

$$\begin{array}{rrcrcrcr}
\min & 3x_1 & + & 2x_2 & + & 4x_3 & & \\
\text{s.t.} & x_1 & + & 2x_2 & - & x_3 & \geq & 6 \\
 & x_1 & - & x_2 & & & \geq & -8 \\
 & 2x_1 & + & x_2 & + & x_3 & = & 9 \\
 & \multicolumn{7}{l}{x_1 \geq 0, \quad x_2 \leq 0, \quad x_3 \text{ urs.}}
\end{array}$$

$$\rightarrow \quad \begin{array}{rrcrcrcr}
\min & 3x_1 & + & 2x_2 & + & 4x_3 & & \\
\text{s.t.} & x_1 & + & 2x_2 & - & x_3 & \geq & 6 \\
 & -x_1 & + & x_2 & & & \leq & 8 \\
 & 2x_1 & + & x_2 & + & x_3 & = & 9 \\
 & \multicolumn{7}{l}{x_1 \geq 0, \quad x_2 \leq 0, \quad x_3 \text{ urs.}}
\end{array}$$

An example

$$
\begin{array}{rrrrrrrrrrr}
 & \min & 3x_1 & - & 2x_2 & + & 4x_3 & - & 4x_4 & & \\
\rightarrow & \text{s.t.} & x_1 & - & 2x_2 & - & x_3 & + & x_4 & \geq & 6 \\
 & & -x_1 & - & x_2 & & & & & \leq & 8 \\
 & & 2x_1 & - & x_2 & + & x_3 & - & x_4 & = & 9 \\
 & & \multicolumn{9}{l}{x_i \geq 0 \quad \forall i = 1, ..., 4}
\end{array}
$$

$$
\begin{array}{rrrrrrrrrrrrrrr}
 & \min & 3x_1 & - & 2x_2 & + & 4x_3 & - & 4x_4 & & & & & & \\
\rightarrow & \text{s.t.} & x_1 & - & 2x_2 & - & x_3 & + & x_4 & - & x_5 & & & = & 6 \\
 & & -x_1 & - & x_2 & & & & & & & + & x_6 & = & 8 \\
 & & 2x_1 & - & x_2 & + & x_3 & - & x_4 & & & & & = & 9 \\
 & & \multicolumn{13}{l}{x_i \geq 0 \quad \forall i = 1, ..., 6.}
\end{array}
$$

Standard form linear programs

- Given **any** linear program, we may find its standard form.
- In general, a standard form linear program can be expressed as

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

- Typically we denote the number of constraints as m and the number of variables as n.
  - So $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^{m \times 1}$, $c \in \mathbb{R}^{1 \times n}$.
  - A is called the **coefficient matrix**.
  - b is called the **RHS vector**.
  - c is called the **objective vector**.
- The objective function can be either max or min.

Standard form linear programs

- The matrix representation is equivalent to

$$
\begin{aligned}
\min \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.} \quad & A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n = b_1 \\
& \qquad\qquad \vdots \\
& A_{i1}x_1 + A_{i2}x_2 + \cdots + A_{in}x_n = b_i \\
& \qquad\qquad \vdots \\
& A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n = b_m \\
& x_j \geq 0 \quad \forall j = 1, ..., n.
\end{aligned}
$$

Standard form linear programs

- If we can solve the standard form LP, we can then construct the solution for the original LP.
- Let's focus on how to solve a standard form linear program.
- We need some preparations, including the definition of basic solutions and basic feasible solutions.

Road map

- Standard form linear programs.
- **Basic solutions**.
- Basic feasible solutions.
- The idea of the simplex method.

Basic solutions

- Consider a standard form LP with m constraints and n variables

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

- We define some special solutions to be **basic solutions**.

Definition 2

A basic solution to a standard form LP is a solution that (1) has $n - m$ variables being equal to 0 and (2) satisfies $Ax = b$.

- The $n - m$ variables chosen to be zero are **nonbasic variables**.
- The remaining m variables, which **may or may not be** zero, are **basic variables**.

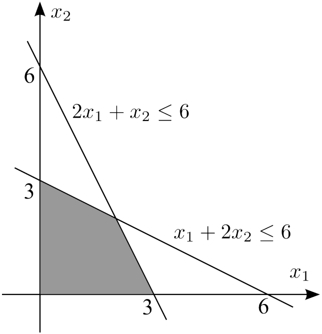
Basic solutions: an example

- Consider an original linear program

$$\begin{array}{rrcrcr} \min & 6x_1 & + & 8x_2 & & \\ \text{s.t.} & x_1 & + & 2x_2 & \leq & 6 \\ & 2x_1 & + & x_2 & \leq & 6 \\ & & x_i \geq 0 & \forall i = 1, 2. & & \end{array}$$

and its standard form

$$\begin{array}{rrcrcrcrcr} \min & 6x_1 & + & 8x_2 & & & & & & \\ \text{s.t.} & x_1 & + & 2x_2 & + & x_3 & & & = & 6 \\ & 2x_1 & + & x_2 & & & + & x_4 & = & 6 \\ & & x_i \geq 0 & \forall i = 1, ..., 4. & & & & & & \end{array}$$



Basic solutions: an example

- In the standard form, $m = 2$ and $n = 4$.
  - There are $n - m = 2$ nonbasic variables.
  - There are $m = 2$ basic variables.
- Steps for obtaining a basic solution:
  - Determine the set of m basic variables, B.
  - The remaining variables form the set of nonbasic variables, N.
  - Set nonbasic variables to zero.
  - Solve the remaining m by m system for the values of basic variables.
- For this example, we will solve a two by two linear system.

Basic solutions: an example

- The two equalities are

$$\begin{array}{rcrcrcrcl} x_1 & + & 2x_2 & + & x_3 & & & = & 6 \\ 2x_1 & + & x_2 & & & + & x_4 & = & 6. \end{array}$$

- Let's try $B = \{x_1, x_2\}$ and $N = \{x_3, x_4\}$:

$$\begin{array}{rcrcl} x_1 & + & 2x_2 & = & 6 \\ 2x_1 & + & x_2 & = & 6. \end{array}$$

  The solution is $(x_1, x_2) = (2, 2)$. Therefore, the basic solution associated with the choice $B = \{x_1, x_2\}$ and $N = \{x_3, x_4\}$ is $(x_1, x_2, x_3, x_4) = (2, 2, 0, 0)$.

Basic solutions: an example

- The two equalities are

$$\begin{array}{rcrcrcrcl} x_1 & + & 2x_2 & + & x_3 & & & = & 6 \\ 2x_1 & + & x_2 & & & + & x_4 & = & 6. \end{array}$$

- Let's try $B = \{x_2, x_3\}$ and $N = \{x_1, x_4\}$:

$$\begin{array}{rcrcl} 2x_2 & + & x_3 & = & 6 \\ x_2 & & & = & 6. \end{array}$$

  The solution is $(x_2, x_3) = (6, -6)$. Therefore, the basic solution associated with the choice $B = \{x_2, x_3\}$ and $N = \{x_1, x_4\}$ is $(x_1, x_2, x_3, x_4) = (0, 6, -6, 0)$.

Basic solutions: an example

- We will call a particular choice of basic variables a **basis**.
  - $\{x_1, x_2\}$ is a basis and $\{x_2, x_3\}$ is another basis.
- Every basic solution is associated with a basis.
- In general, as we need to choose m out of n variables to be basic, we have $\binom{n}{m}$ different bases.
- In this example, we have $\binom{4}{2} = 6$ bases.

Bases

- All the six bases and associated basic variables are listed below:

Basis	Basic solution			
	x_1	x_2	x_3	x_4
$\{x_1, x_2\}$	2	2	0	0
$\{x_1, x_3\}$	3	0	3	0
$\{x_1, x_4\}$	6	0	0	-6
$\{x_2, x_3\}$	0	6	-6	0
$\{x_2, x_4\}$	0	3	0	3
$\{x_3, x_4\}$	0	0	6	6

- Basic variables have nothing to do with the objective function!

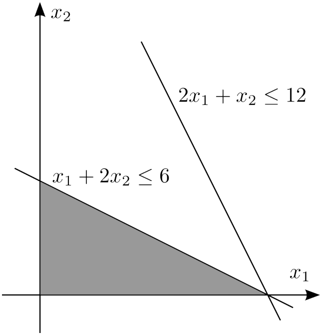
Basic solutions v.s. bases

- For a basis, what matters are **variables**, not **values**.
- Consider another example

$$\begin{array}{rrcrcr} \min & 6x_1 & + & 8x_2 & & \\ \text{s.t.} & x_1 & + & 2x_2 & \leq & 6 \\ & 2x_1 & + & x_2 & \leq & 12 \\ & & x_i \geq 0 & \forall i = 1, 2 & & \end{array}$$

and its standard form

$$\begin{array}{rrcrcrcrcr} \min & 6x_1 & + & 8x_2 & & & & & & \\ \text{s.t.} & x_1 & + & 2x_2 & + & x_3 & & & = & 6 \\ & 2x_1 & + & x_2 & & & + & x_4 & = & 12 \\ & x_i \geq 0 & \forall i = 1, ..., 4. & & & & & & & \end{array}$$



Basic solutions v.s. bases

- The six bases and the associated basic variables are listed below:

Basis	Basic solution			
	x_1	x_2	x_3	x_4
$\{x_1, x_2\}$	**6**	**0**	0	0
$\{x_1, x_3\}$	**6**	0	**0**	0
$\{x_1, x_4\}$	**6**	0	0	**0**
$\{x_2, x_3\}$	0	12	-18	0
$\{x_2, x_4\}$	0	3	0	9
$\{x_3, x_4\}$	0	0	6	12

- Three different bases result in **the same** basic solution!

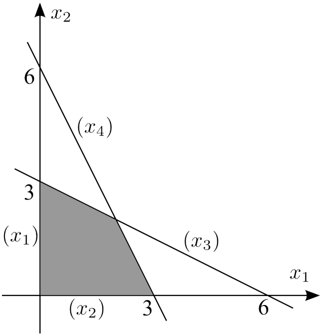
Basic solutions v.s. bases

- In general, **multiple bases** may be mapped to a **single basic solution**.
  - This happens if and only if at least one basic variable is (coincidentally) 0.
- For n variables and m equalities, there are always **exactly** $\binom{n}{m}$ bases and **at most** $\binom{n}{m}$ distinct basic solutions.
- When multiple bases correspond to one single basic solution, the linear program is **degenerate**.
- When may this happen?
  - To answer this question, we need to study the relationship between variables and constraints first.

Original and slack variables

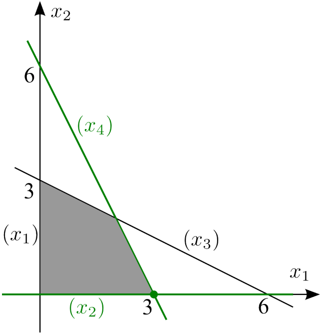
- Among all variables of a standard form LP, some are **original** while some are **slack**.
  - Each original variable corresponds to a **nonnegative** constraint.
  - Each slack variable corresponds to a **functional** constraint.

$$\begin{array}{rrcrcr} \min & 6x_1 & + & 8x_2 & & \\ \text{s.t.} & x_1 & + & 2x_2 & \leq & 6 \\ & 2x_1 & + & x_2 & \leq & 6 \\ & & & x_i \geq 0 & \forall i = 1, 2 & \end{array}$$



Nonbasic variables vs. binding constraints

- Each basis corresponds to a set of m **binding constraints**.
  - When an **original** variable is nonbasic, it becomes 0 and the corresponding **nonnegative** constraint is binding.
  - When a **slack** variable is nonbasic, it becomes 0 and the corresponding **functional** constraint is binding.
- E.g., for the basis $\{x_1, x_3\}$, the constraints $x_2 \geq 0$ and $2x_1 + x_2 \leq 6$ are binding.

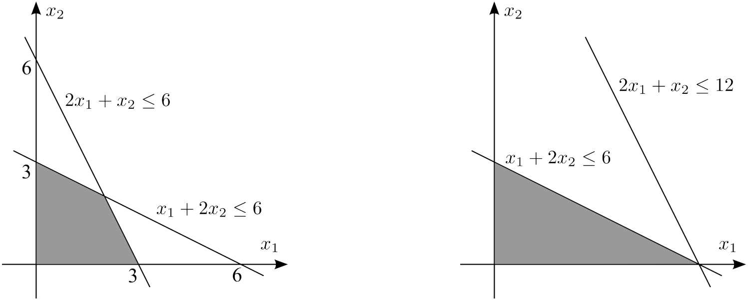


When is an LP degenerate?

- An LP is degenerate when multiple bases correspond to one single basic solution.
  - A basis
    ⇔ a set of nonbasic variables
    ⇔ a set of binding constraints
    ⇔ an intersection of these constraints.
  - More than $n - m$ constraints intersect at one single point
    ⇔ Multiple ways of choosing $n - m$ binding constraints at a point
    ⇔ Multiple bases correspond to this point
    ⇔ Multiple bases correspond to the same basic solution
    ⇔ Degenerate LP.

When is an LP degenerate?

- More than $n - m$ constraints intersect at one single point.
  - $n = 4$, $m = 2$; we are talking about the standard form!



- How to illustrate this situation in a three-dimensional space?

Degeneracy of linear programs

- **Degeneracy** may cause severe problems in solving linear programs.
  - It hurts computational **efficiency**.
  - Especially when using the simplex method.
- Nevertheless, let's skip this issue and consider **nondegenerate linear programs** first.
- In other words, we will assume that different bases correspond to different basic solutions.

Road map

- Standard form linear programs.
- Basic solutions.
- **Basic feasible solutions**.
- The idea of the simplex method.

Basic feasible solutions

- Among all basic solutions, some are feasible.
  - By the definition of basic solutions, they satisfy $Ax = b$.
  - If one also satisfies $x \geq 0$, it satisfies all constraints.
- In this case, it is called **basic feasible solutions** (bfs).

Definition 3 (Basic feasible solution)

A basic feasible solution to a standard form LP is a basic solution whose basic variables are all nonnegative.

- We do not need to restrict the values of nonbasic variables. Why?

Basic feasible solutions and extreme points

- We may link extreme points and basic feasible solutions:

Proposition 1 (Extreme points and basic feasible solutions)

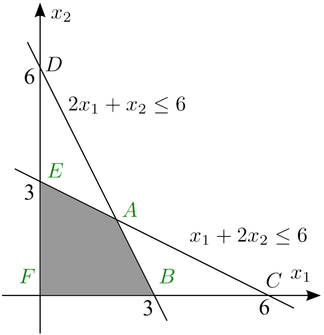
For a standard form LP, a solution is an extreme point of the feasible region if and only if it is a basic feasible solution to the LP.

Proof. Beyond the scope of this course. □

- Intuition: An extreme point is feasible. Also, it locates at a "corner", which is the intersection of at least $n - m$ constraints, so it is a basic solution.

Basic feasible solutions and extreme points

Basis	Bfs?	Point	Basic solution			
			x_1	x_2	x_3	x_4
$\{x_1, x_2\}$	Yes	A	2	2	0	0
$\{x_1, x_3\}$	Yes	B	3	0	3	0
$\{x_1, x_4\}$	No	C	6	0	0	-6
$\{x_2, x_3\}$	No	D	0	6	-6	0
$\{x_2, x_4\}$	Yes	E	0	3	0	3
$\{x_3, x_4\}$	Yes	F	0	0	6	6



Basic feasible solutions

- What's the implication of the previous proposition?

Proposition 2 (Optimality of basic feasible solutions)

For a standard form LP, if there is an optimal solution, there is an optimal basic feasible solution.

Proof. We know there is a one-to-one mapping between extreme points and basic feasible solutions. Moreover, we know if there is an optimal solution, there is an optimal extreme point solution. The proof then follows. □

Basic feasible solutions vs. extreme points

- To find an optimal solution:
  - Instead of searching among all extreme points, we may search among **all basic feasible solutions**.
  - But the two sets are equally large! What is the difference?
- Given a solution:
  - Checking whether it is a basic feasible solution is easy: just count **the number of zeros** and verify **nonnegativity**.
  - Checking whether it is an extreme point is hard (for computers).
- Given a linear program:
  - **Enumerating** all basic feasible solutions is possible.
  - How to enumerate all extreme points?

Basic feasible solutions

- Listing all basic feasible solutions are possible but **unrealistic**.
  - For a linear program with n variables and m constraints, we have $\binom{n}{m}$ bases and thus at most $\binom{n}{m}$ basic feasible solutions. There are **too many** to list in a reasonable time!
- The simplex method is a "**smart**" way of searching among all basic feasible solutions.
- Its idea is to improve a current basic feasible solution by moving to a better basic feasible solution.
- Let's define **adjacent** basic feasible solutions first.

Adjacent basic feasible solutions

- Two basic feasible solutions may or may not be **adjacent**:

Definition 4 (Adjacent basic feasible solutions)

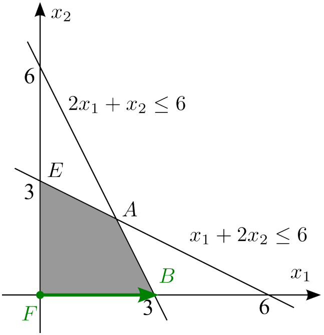
Two bases are adjacent if exactly one of their variable is different. Two basic feasible solutions are adjacent if their associated bases are adjacent.

  - $\{x_1, x_2\}$ and $\{x_1, x_4\}$ are adjacent.
  - $\{x_1, x_2\}$ and $\{x_3, x_4\}$ are not adjacent.
  - How about $\{x_1, x_2\}$ and $\{x_2, x_4\}$?

Adjacent basic feasible solutions

- A pair of adjacent basic feasible solutions correspond to a pair of "adjacent" extreme points.
  - Extreme points that are on **the same edge**.
- Moving from a bfs to its adjacent bfs is **moving along an edge**.

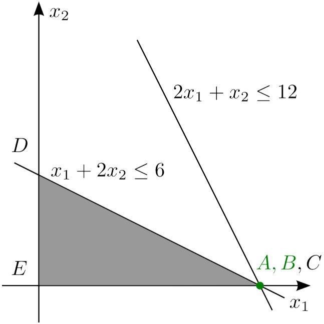
Basis	Point	Basic solution			
		x_1	x_2	x_3	x_4
$\{x_1, x_2\}$	A	2	2	0	0
$\{x_1, x_3\}$	B	3	0	3	0
$\{x_2, x_4\}$	E	0	3	0	3
$\{x_3, x_4\}$	F	0	0	6	6



Adjacent basic feasible solutions

- Adjacency is defined based on **variables**, not values!
  - Points A and B are the same point, but bases $\{x_1, x_2\}$ and $\{x_1, x_3\}$ are adjacent, even though no value is different.
  - With degeneracy, **adjacent bfs** may be actually **identical**.

Basis	Point	Basic solution			
		x_1	x_2	x_3	x_4
$\{x_1, x_2\}$	A	**6**	**0**	0	0
$\{x_1, x_3\}$	B	**6**	0	**0**	0
$\{x_1, x_4\}$	C	6	0	0	0
$\{x_2, x_3\}$	D	0	3	0	9
$\{x_2, x_4\}$	E	0	0	6	12



Moving to the next basic feasible solutions

- Imagine that you are currently at one basic feasible solution.
  - Let's call it $x^1 = (x_1^1, x_2^1, ..., x_n^1)$.
- You want to move to a **better** basic feasible solution.
  - Let's call the new basic feasible solution $x^2 = (x_1^2, x_2^2, ..., x_n^2)$.
  - We want $cx^2 < cx^1$ when we want to minimize cx.
- How many different x^2 do we need to examine?
  - Among m basic variables, we choose one to **leave** the basis.
  - Among $n - m$ nonbasic variables, we choose one to **enter** the basis.
  - In total we have $m(n - m)$ candidates.
- How to choose one? The simplex method!

Road map

- Standard form linear programs.
- Basic solutions.
- Basic feasible solutions.
- **The idea of the simplex method**.

The simplex method

- Below we will describe the main idea of the simplex method for solving standard form linear programs.
- All we need is to search among basic feasible solutions.
- Suppose we are standing on a bfs x^1. We want to move to an adjacent bfs x^2. We need to
  - select one **nonbasic** variable to **enter** the basis, and
  - select one **basic** variable to **leave** the basis.

The entering variable

- Selecting one nonbasic variable to enter means making it **nonzero**.
  - If it is an original variable, we leave the associated axis.
  - If it is a slack variable, we leave the associated functional constraint.
  - In short, one constraint becomes **nonbinding**.
  - We will move **along the edge** that leaves the constraint.
- For a linear program, we may simply choose a direction that **improves** the current solution.
  - Why?
  - Because "a local optimum is a global optimum."

The entering variable

- Consider the linear program

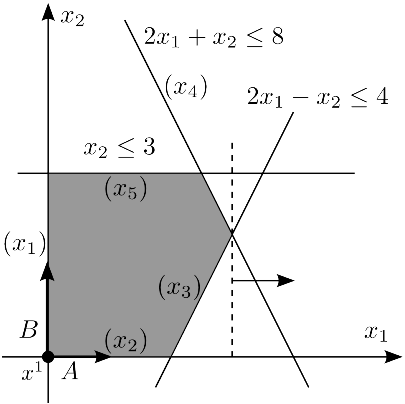
$$\begin{array}{lccccc} \min & -x_1 & & & & \\ \text{s.t.} & 2x_1 & - & x_2 & \leq & 4 \\ & 2x_1 & + & x_2 & \leq & 8 \\ & & & x_2 & \leq & 3 \\ & \multicolumn{5}{c}{x_i \geq 0 \quad \forall\, i = 1, 2.} \end{array}$$

and its standard form

$$\begin{array}{lccccccccccc} \min & -x_1 & & & & & & & & & & \\ \text{s.t.} & 2x_1 & - & x_2 & + & x_3 & & & & & = & 4 \\ & 2x_1 & + & x_2 & & & + & x_4 & & & = & 8 \\ & & & x_2 & & & & & + & x_5 & = & 3 \\ & \multicolumn{11}{l}{x_i \geq 0 \quad \forall\, i = 1, ..., 5.} \end{array}$$

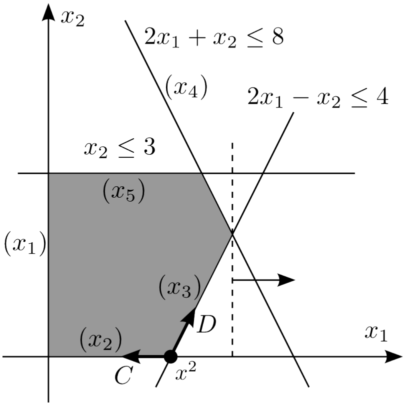
The entering variable

- For the bfs x^1:
  - The basis is $\{x_3, x_4, x_5\}$.
  - x_1 and x_2 are nonbasic.
  - Let x_1 enters $\Rightarrow$ makes $x_1 > 0 \Rightarrow$ move along direction A, constraint $x_2 \geq 0$.
  - Let x_2 enters $\Rightarrow$ move along direction B, constraint $x_1 \geq 0$.



The entering variable

- For the bfs x^2:
  - The basis is $\{x_1, x_4, x_5\}$.
  - x_2 and x_3 are nonbasic.
  - Let x_2 enters $\Rightarrow$ makes $x_2 > 0 \Rightarrow$ move along direction D, constraint $2x_1 - x_2 \leq 4$.
  - Let x_3 enters $\Rightarrow$ move along direction C, constraint $x_2 \geq 0$.

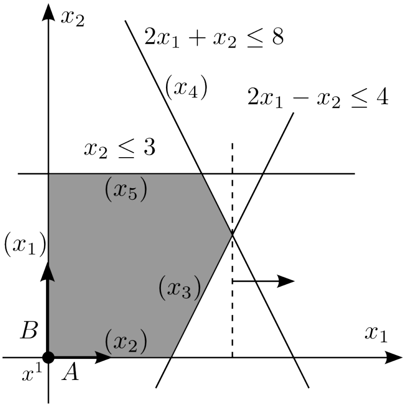


The leaving variable

- Suppose we have chosen one entering variable.
  - We have chosen one improving direction to go.
- How to choose a **leaving** variable?
  - When should we **stop**?
- We should stop when we "**hit a constraint**", i.e., when one basic variable becomes 0.
  - This basic variable will leave the basis.
  - As it becomes 0, it becomes a nonbasic variable.

The leaving variable

- For the bfs x^1, suppose we move along direction A.
  - The original basis is $\{x_3, x_4, x_5\}$.
  - x_1 **enters** the basis.
  - We **first hit** $2x_1 - x_2 \leq 4$.
  - x_3 becomes 0.
  - x_3 becomes nonbasic.
  - x_3 **leaves** the basis.
  - The new basis becomes $\{x_1, x_4, x_5\}$.



An iteration

- At a basic feasible solution, we move to another **better** basic feasible solution.
  - We first choose **which direction to go** (the **entering** variable). That will be an improving direction along an edge.
  - We then determine **when to stop** (the **leaving** variable). That depends on the first constraint we hit.
  - We may then treat the new bfs as the current bfs and then **repeat**.
- We stop when there is no direction to go (no improving direction).
- The process of moving to the next bfs is call an **iteration**.

The simplex method

- The simplex method is simple:
  - It suffices to **move along edges** (because we only need to search among extreme points).
  - At each point, the number of directions to search for is **small** (because we consider only edges).
  - For each improving direction, the **stopping condition** is simple: Keep moving forwards until we cannot.
- The simplex method is smart:
  - When at a point there is **no improving direction** along an edge, we may claim that the point is optimal.