Introduction to Operations Research

Deterministic Models

JURAJ STACHO

Department of Industrial Engineering and Operations Research

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Contents

Preface

These lecture notes were written during the Fall/Spring 2013/14 semesters to accompany lectures of the course *IEOR 4004: Introduction to Operations Research - Deterministic Models*. The notes were meant to provide a succint summary of the material, most of which was loosely based on the book *Winston-Venkataramanan: Introduction to Mathematical Programming (4th ed.), Brooks/Cole 2003*. Other material (such as the dictionary notation) was adapted from *Chvátal: Linear Programming, Freeman 1983* and *Dantzig-Thapa: Linear Programming, Springer-Verlag 1997*. Various other bits were inspired by other lecture notes and sources on the Internet. These notes are not meant to replace any book; interested reader will find more details and examples in the Winston book in particular. I would like to thank students that helped me correct numerous mistakes in the earlier versions of the notes. Most likely all mistakes have not been yet caught, and so the reader should exercise caution should there be inconsistencies in the text. I am passing on the notes to Prof. Strickland who will continue making adjustments to the material as needed for the upcoming offerings of the course. Of course, any suggestions for improvements are welcomed from anyone interested.

Juraj Stacho
July 26, 2014

1 Mathematical modeling by example

Product mix

A toy company makes two types of toys: *toy soldiers* and *trains*. Each toy is produced in two stages, first it is constructed in a carpentry shop, and then it is sent to a finishing shop, where it is varnished, vaxed, and polished. To make one toy soldier costs \$10 for raw materials and \$14 for labor; it takes 1 hour in the carpentry shop, and 2 hours for finishing. To make one train costs \$9 for raw materials and \$10 for labor; it takes 1 hour in the carpentry shop, and 1 hour for finishing.

There are 80 hours available each week in the carpentry shop, and 100 hours for finishing. Each toy soldier is sold for \$27 while each train for \$21. Due to decreased demand for toy soldiers, the company plans to make and sell at most 40 toy soldiers; the number of trains is not restriced in any way.

What is the optimum (*best*) product mix (i.e., what quantities of which products to make) that *maximizes* the profit (assuming all toys produced will be sold)?

Terminology

decision variables:	$x_1, x_2, \ldots, x_i, \ldots$
variable domains: values that variables can take	$x_1, x_2 \geq 0$
goal/objective:	maximize/minimize
objective function: function to minimize/maximize	$2x_1 + 5x_2$
constraints: equations/inequalities	$3x_1 + 2x_2 \leq 10$

Example

Decision variables:

- x_1= # of toy soldiers
- x_2= # of toy trains

Objective: maximize profit

- $\$27 - \$10 - \$14 = \$3$ profit for selling one toy soldier $\Rightarrow 3x_1$ profit (in \$) for selling x_1 toy soldier
- $\$21 - \$9 - \$10 = \$2$ profit for selling one toy train $\Rightarrow 2x_2$ profit (in \$) for selling x_2 toy train

$\Rightarrow \underbrace{z = 3x_1 + 2x_2}$ profit for selling x_1 toy soldiers and x_2 toy trains

objective function

Constraints:

- producing x_1 toy soldiers and x_2 toy trains requires
  (a) $1x_1 + 1x_2$ hours in the carpentry shop; there are 80 hours available
  (b) $2x_1 + 1x_2$ hours in the finishing shop; there are 100 hours available
- the number x_1 of toy soldiers produced should be at most 40

Variable domains: the numbers x_1, x_2 of toy soldiers and trains must be non-negative (**sign restriction**)

$$\begin{array}{rrcl} \text{Max} & 3x_1 + 2x_2 & & \\ & x_1 + x_2 & \leq & 80 \\ & 2x_1 + x_2 & \leq & 100 \\ & x_1 & \leq & 40 \\ & x_1, x_2 & \geq & 0 \end{array}$$

We call this a **program**. It is a **linear** program, because the objective is a linear function of the decision variables, and the constraints are linear inequalities (in the decision variables).

Blending

A company wants to produce a certain alloy containing 30% lead, 30% zinc, and 40% tin. This is to be done by mixing certain amounts of existing alloys that can be purchased at certain prices. The company wishes to minimize the cost. There are 9 available alloys with the following composition and prices.

Alloy	1	2	3	4	5	6	7	8	9	Blend
Lead (%)	20	50	30	30	30	60	40	10	10	30
Zinc (%)	30	40	20	40	30	30	50	30	10	30
Tin (%)	50	10	50	30	40	10	10	60	80	40
Cost ($/lb)	7.3	6.9	7.3	7.5	7.6	6.0	5.8	4.3	4.1	minimize

Designate a *decision* variables $x_1, x_2, \ldots, x_9$ where

x_i is the **amount** of Alloy i in a **unit of blend**

In particular, the decision variables must satisfy $x_1 + x_2 + \ldots + x_9 = 1$. (It is a common mistake to choose x_i the **absolute** amount of Alloy i in the blend. That may lead to a non-linear program.)

With that we can setup constraints and the objective function.

$$\begin{array}{lrcl} \text{Min} & 7.3x_1 + 6.9x_2 + 7.3x_3 + 7.5x_4 + 7.6x_5 + 6.0x_6 + 5.8x_7 + 4.3x_8 + 4.1x_9 & = & z \quad \text{[Cost]} \\ \text{s.t.} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 & = & 1 \\ & 0.2x_1 + 0.5x_2 + 0.3x_3 + 0.3x_4 + 0.3x_5 + 0.6x_6 + 0.4x_7 + 0.1x_8 + 0.1x_9 & = & 0.3 \quad \text{[Lead]} \\ & 0.3x_1 + 0.4x_2 + 0.2x_3 + 0.4x_4 + 0.3x_5 + 0.3x_6 + 0.5x_7 + 0.3x_8 + 0.1x_9 & = & 0.3 \quad \text{[Zinc]} \\ & 0.5x_1 + 0.1x_2 + 0.5x_3 + 0.3x_4 + 0.4x_5 + 0.1x_6 + 0.1x_7 + 0.6x_8 + 0.8x_9 & = & 0.4 \quad \text{[Tin]} \end{array}$$

Do we need **all** the four equations?

Product mix (once again)

Furniture company manufactures four models of chairs. Each chair requires certain amount of raw materials (wood/steel) to make. The company wants to decide on a production that maximizes profit (assuming all produced chair are sold). The required and available amounts of materials are as follows.

	Chair 1	Chair 2	Chair 3	Chair 4	Total available
Steel	1	1	3	9	4,4000 (lbs)
Wood	4	9	7	2	6,000 (lbs)
Profit	\$12	\$20	\$18	\$40	maximize

Decision variables:

$x_i =$ the number of chairs of type i produced
each x_i is non-negative

Objective function:

maximize profit $z = 12x_1 + 20x_2 + 18x_3 + 40x_4$

Costraints:

at most $4,400$ lbs of steel available: $x_1 + x_2 + 3x_3 + 9x_4 \leq 4,400$
at most $6,000$ lbs of wood available: $4x_1 + 9x_2 + 7x_3 + 2x_4 \leq 6,000$

Resulting program:

$$\begin{array}{lrcrcrcrcll} \text{Max} & 12x_1 & + & 20x_2 & + & 18x_3 & + & 40x_4 & = & z & \text{[Profit]} \\ \text{s.t.} & x_1 & + & x_2 & + & 3x_3 & + & 9x_4 & \leq & 4,400 & \text{[Steel]} \\ & 4x_1 & + & 9x_2 & + & 7x_3 & + & 2x_4 & \leq & 6,000 & \text{[Wood]} \end{array}$$

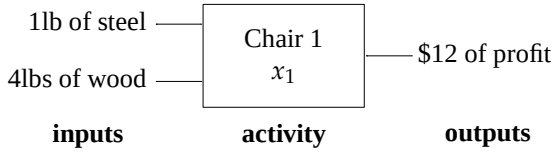
$$x_1, x_2, x_3, x_4 \geq 0$$

1.1 Activity-based formulation

Instead of constructing the formulation as before (row-by-row), we can proceed by columns.

We can view columns of the program as **activities**. An activity has

inputs: materials consumed per unit of activity	(1lb of steel and 4lbs of wood)
outputs: products produced per unit of activity	(\$12 of profit)
activity level: a level at which we operate the activity	(indicated by a variable x_1)



Operating the activity "Chair 1" at level x_1 means that we produce x_1 chairs of type 1, each consuming 1lb of steel, 4lbs of wood, and producing \$12 of profit. Activity levels are always assumed to be **non-negative**.

The materials/labor/profit consumed or produced by an activity are called **items** (correspond to rows).

The effect of an activity on items (i.e. the amounts of items that are consumed/produced by an activity) are **input-output coefficients**.

The total amount of items available/supplied/required is called the **external flow of items**.

We choose **objective** to be one of the items which we choose to maximize or minimize.

Last step is to write **material balance equations** that express the flow of items in/out of activies and with respect to the external flow.

Example

Items: Steel
Wood
Profit

External flow of items:
Steel: 4,400lbs of available (flowing in)
Wood: 6,000lbs of available (flowing in)

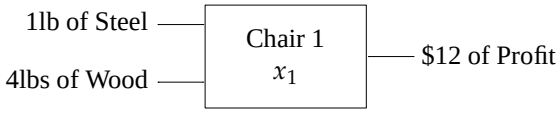
Objective:

Profit: maximize (flowing out)

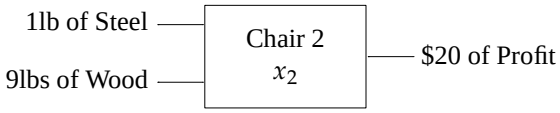
Activities:

producing a chair of type i where $i = 1, 2, 3, 4$, each is assigned an activity level x_i

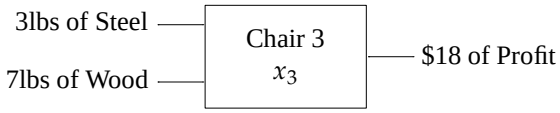
Chair 1: Producing 1 chair of type 1
consumes 1 lb of Steel
4 lbs of Wood
produces \$12 of Profit



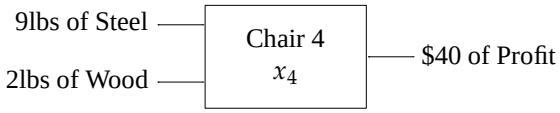
Chair 2: Producing 1 chair of type 2
consumes 1 lb of Steel
9 lbs of Wood
produces \$20 of Profit



Chair 3: Producing 1 chair of type 3
consumes 3 lbs of Steel
7 lbs of Wood
produces \$18 of Profit

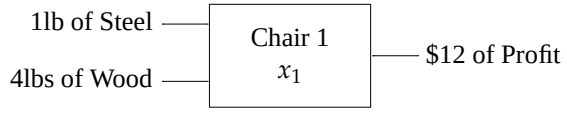


Chair 4: Producing 1 chair of type 4
consumes 9 lbs of Steel
2 lbs of Wood
produces \$40 of Profit



The material balance equations:

To see how to do this, consider activity Chair 1: consumes 1lb of Steel, 4lbs of Wood, and produces \$12 of Profit. Thus at level x_1, we consume $1x_1$ lbs of Steel, $4x_1$ lbs of Wood, and produce $12x_1$ dollars of Profit.



$$\begin{aligned} \ldots + 12x_1 + \ldots \quad & [\text{Profit}] \\ \ldots + 1x_1 + \ldots \quad & [\text{Steel}] \\ \ldots + 4x_1 + \ldots \quad & [\text{Wood}] \end{aligned}$$

On the right, you see the effect of operating the activity at level x_1. (Note in general we will adopt a different **sign convention**; we shall discuss is in a later example.)

Thus considering all activities we obtain:

$$\begin{aligned} 12x_1 + 20x_2 + 18x_3 + 40x_4 \quad & [\text{Profit}] \\ x_1 + x_2 + 3x_3 + 9x_4 \quad & [\text{Steel}] \\ 4x_1 + 9x_2 + 7x_3 + 2x_4 \quad & [\text{Wood}] \end{aligned}$$

Finally, we incorporate the external flow and objective: 4,400lbs of Steel available, $6,000$lbs of Wood available, maximize profit:

$$\begin{aligned} \text{Max} \quad & 12x_1 + 20x_2 + 18x_3 + 40x_4 = z && [\text{Profit}] \\ \text{s.t.} \quad & x_1 + x_2 + 3x_3 + 9x_4 \leq 4,400 && [\text{Steel}] \\ & 4x_1 + 9x_2 + 7x_3 + 2x_4 \leq 6,000 && [\text{Wood}] \end{aligned}$$

$$x_1, x_2, x_3, x_4 \geq 0$$

2 Linear Programming

Linear program (LP) in a **standard form** (maximization)

$$
\begin{array}{lccccccccll}
\max & c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n & & & \text{Objective function} \\
\text{subject to} & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \leq & b_1 & \left.\begin{array}{l}\\\\\\\\\end{array}\right\} \\
& a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \leq & b_2 & \\
& \vdots & + & \vdots & & & & \vdots & & \vdots & \text{Constraints} \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \leq & b_m & \left.\begin{array}{l}\\\\\\\\\end{array}\right\} \\
& & & & & & & x_1, x_2, \dots, x_n & \geq & 0 & \text{Sign restrictions}
\end{array}
$$

Feasible solution (point) $P = (p_1, p_2, \dots, p_n)$ is an assignment of values to the $p_1, \dots, p_n$ to variables $x_1, \dots, x_n$ that satisfies **all** constraints and **all** sign restrictions.

Feasible region $\equiv$ the set of all feasible points.

Optimal solution $\equiv$ a feasible solution with maximum value of the objective function.

2.1 Formulating a linear program

1. Choose decision variables
2. Choose an objective and an objective function – linear function in variables
3. Choose constraints – linear inequalities
4. Choose sign restrictions

Example

You have \$100. You can make the following three types of investments:

Investment A. Every dollar invested now yields \$0.10 a year from now, and \$1.30 three years from now.

Investment B. Every dollar invested now yields \$0.20 a year from now and \$1.10 two years from now.

Investment C. Every dollar invested a year from now yields \$1.50 three years from now.

During each year leftover cash can be placed into money markets which yield 6% a year. The most that can be invested a single investment (A, B, or C) is \$50.

Formulate an LP to maximize the available cash three years from now.

Decision variables: x_A, x_B, x_C, amounts invested into Investments A, B, C, respectively
y_0, y_1, y_2, y_3 cash available/invested into money markets now, and in 1,2,3 years.

$$
\begin{array}{lrcrcrcrcr}
\text{Max} & y_3 & & & & & & & & \\
\text{s.t.} & x_A & + & x_B & & & + & y_0 & = & 100 \\
 & 0.1x_A & + & 0.2x_B & - & x_C & + & 1.06y_0 & = & y_1 \\
 & & & 1.1x_B & & & + & 1.06y_1 & = & y_2 \\
 & 1.3x_A & & & + & 1.5x_C & + & 1.06y_2 & = & y_3 \\
 & x_A & & & & & & & \leq & 50 \\
 & & & x_B & & & & & \leq & 50 \\
 & & & & & x_C & & & \leq & 50 \\
 & & & & & x_A, x_B, x_C, y_0, y_1, y_2, y_3 & & & \geq & 0
\end{array}
$$

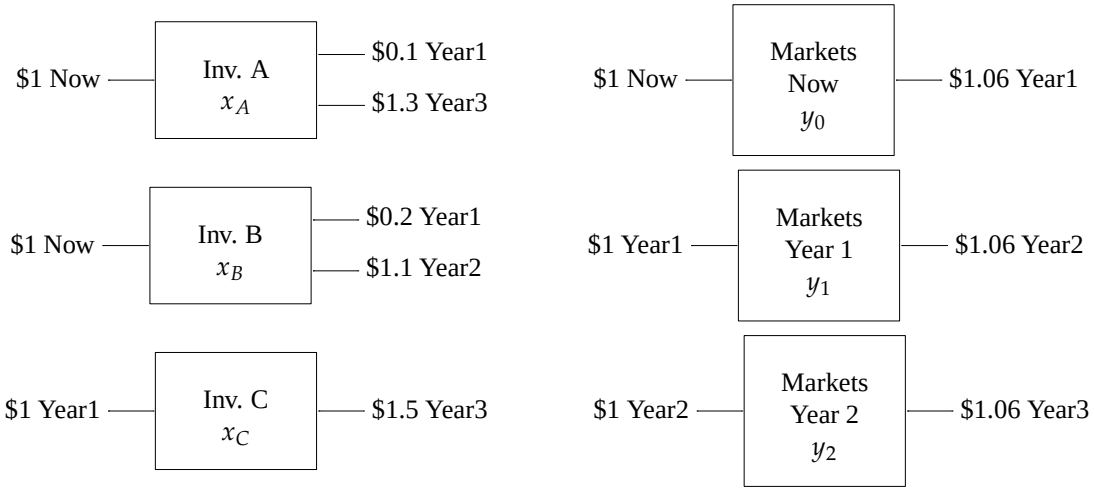
Activities

Items		Inv. A	Inv. B	Inv. C	Markets Now	Markets Year 1	Markets Year 2		External flow
	Now	-1	-1		-1			=	-100
	Year1	0.1	0.2	-1	1.06	-1		=	0
	Year2		1.1			1.06	-1	=	0
	Year3	1.3		1.5			1.06		maximize

Sign convention: inputs have **negative** sign, outputs have **positive** signs.

External in-flow has **negative** sign, external out-flow has **positive** sign.

We have in-flow of \$100 cash "Now" which means we have $-$\$100 on the right-hand side. No in-flow or out-flow of any other item.



$$
\begin{array}{lrcrcrcrcr}
\text{Max} & 1.3x_A & & & + & 1.5x_C & & & + & 1.06y_2 \\
\text{s.t.} & x_A & + & x_B & & & + & y_0 & & = 100 \\
 & 0.1x_A & + & 0.2x_B & - & x_C & + & 1.06y_0 - & y_1 & = 0 \\
 & & & 1.1x_B & & & & + 1.06y_1 - & y_2 & = 0 \\
 & & & & & & & y_0, y_1, y_2 & & \geq 0 \\
 & & & & & & & 0 \leq x_A, x_B, x_C & & \leq 50
\end{array}
$$

Post office problem

Post office requires different numbers of full-time employees on different days. Each full time employee works 5 consecutive days (e.g. an employee may work from Monday to Friday or, say from Wednesday to Sunday). Post office wants to hire minimum number of employees that meet its daily requirements, which are as follows.

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
17	13	15	19	14	16	11

Let x_i denote the number of employees that **start working** in day i where $i = 1, ..., 7$ and work for 5 consecutive days from that day. How many workers work on Monday? Those that start on Monday, or Thursday, Friday, Saturday, or Sunday. Thus $x_1 + x_4 + x_5 + x_6 + x_7$ should be at least 17.

Then the formulation is thus as follows:

$$
\begin{array}{lccccccccccccccc}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & & \\
\text{s.t.} & x_1 & & & & & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \geq & 17 \\
 & x_1 & + & x_2 & & & & & + & x_5 & + & x_6 & + & x_7 & \geq & 13 \\
 & x_1 & + & x_2 & + & x_3 & & & & & + & x_6 & + & x_7 & \geq & 15 \\
 & x_1 & + & x_2 & + & x_3 & + & x_4 & & & & & + & x_7 & \geq & 19 \\
 & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & & & & & \geq & 14 \\
 & & & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & & & \geq & 16 \\
 & & & & & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \geq & 11 \\
 & & & & & & & & & & & x_1, x_2, \ldots, x_7 & & & \geq & 0
\end{array}
$$

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday		
Total	1	1	1	1	1	1	1		minimize
Monday	1			1	1	1	1	$\geq$	17
Tuesday	1	1			1	1	1	$\geq$	13
Wednesday	1	1	1			1	1	$\geq$	15
Thursday	1	1	1	1			1	$\geq$	19
Friday	1	1	1	1	1			$\geq$	14
Saturday		1	1	1	1	1		$\geq$	16
Sunday			1	1	1	1	1	$\geq$	11

(Simple) Linear regression

Given a set of datapoints $\{(1,2),(3,4),(4,7)\}$ we want to find a line that most closely represents the datapoints. There are various ways to measure what it means "closely represent". We may, for instance, minimize the average distance (deviation) of the datapoints from the line, or minimize the sum of distances, or the sum of squares of distances, or minimize the maximum distance of a datapoint from the line. Here the distance can be either Euclidean distance, or vertical distance, or Manhattan distance (vertical+horizontal), or other.

We choose to minimize the maximum vertical distance of a point from the line. A general equation of a line with finite slope has form $y = ax + c$ where a and c are parameters. For a point (p,q), the vertical distance of the point from the line $y = ax + c$ can be written as $|q - ap - c|$. Thus we want

Problem: Find constants a, c such that the largest of the three values $|2 - a - c|$, $|4 - 3a - c|$, $|7 - 4a - c|$ is as small as possible.

$$\min \max \left\{ |2 - a - c|,\ |4 - 3a - c|,\ |7 - 4a - c| \right\}$$

We want to formulate it as a linear program. Issues: non-negativity, the absolute value, the min of max.

- the min of max: $w \geq \max\{i_1, i_2, \ldots, i_t\}$ if and only if $w \geq i_1$ and $w \geq i_2$ and $\ldots$ and $w \geq i_t$

$$\begin{array}{ll} \text{Min} & w \\ \text{s.t.} & w \geq |2 - 1a - c| \\ & w \geq |4 - 3a - c| \\ & w \geq |7 - 4a - c| \end{array}$$

- absolute values: $w \geq |i|$ if and only if $w \geq i$ and $w \geq -i$.

  (in other words, the absolute value of i is at most w if and only if $-w \leq i \leq w$)

$$\begin{array}{ll} \text{Min} & w \\ \text{s.t.} & w \geq 2 - a - c \\ & w \geq -2 + a + c \\ & w \geq 4 - 3a - c \\ & w \geq -4 + 3a + c \\ & w \geq 7 - 4a - c \\ & w \geq -7 + 4a + c \end{array}$$

- unrestricted sign: write $x = x^+ - x^-$ where $x^+, x^- \geq 0$ are new variables

$$\begin{array}{ll} \text{Min} & w \\ \text{s.t.} & w \geq 2 - a^+ + a^- - c^+ + c^- \\ & w \geq -2 + a^+ - a^- + c^+ - c^- \\ & w \geq 4 - 3a^+ + 3a^- - c^+ + c^- \\ & w \geq -4 + 3a^+ - 3a^- + c^+ - c^- \\ & w \geq 7 - 4a^+ + 4a^- - c^+ + c^- \\ & w \geq -7 + 4a^+ - 4a^- + c^+ - c^- \\ & \text{where } a^+, a^-, c^+, c^-, w \geq 0 \end{array}$$

$$\begin{array}{ll} \text{Min} & w \\ \text{s.t.} & w + a^+ - a^- + c^+ - c^- \geq 2 \\ & w - a^+ + a^- - c^+ + c^- \geq -2 \\ & w + 3a^+ - 3a^- + c^+ - c^- \geq 4 \\ & w - 3a^+ + 3a^- - c^+ + c^- \geq -4 \\ & w + 4a^+ - 4a^- + c^+ - c^- \geq 7 \\ & w - 4a^+ + 4a^- - c^+ + c^- \geq -7 \\ & a^+, a^-, c^+, c^-, w \geq 0 \end{array}$$

Note

The above formulation on the right is **standard** form of a minimization LP. We have already seen the standard form of a maximization problem; this is the same except that we minimize the objective function and the signs of inequalities switch (this is only done for convenience sake when we get to solving LPs).

2.2 Summary and further tricks

Let us summarize what we have learned so far.

- Linear Program (LP) is an optimization problem where
  - → the **goal** is to maximize or minimize a **linear objective function**
  - → over a set of **feasible solutions** – i.e. solution of a set of **linear inequalities** (forming the **feasible region**).

- **Standard form**: all inequalities are $\leq$-inequalities (or all are $\geq$-inequalities) and all variables are non-negative
  - → to get a $\leq$-inequality from a $\geq$-inequality we multiply both sides by -1 and reverse the sign (this gives us an equivalent problem)

$$x_1 - x_2 \leq 100 \qquad \Longleftrightarrow \qquad -x_1 + x_2 \geq -100$$

  - → to get inequalities from an equation, we replace it by two identical inequalities, one with $\leq$ and one with $\geq$

$$x_1 - x_2 = 100 \qquad \Longleftrightarrow \qquad \begin{array}{l} x_1 - x_2 \leq 100 \\ x_1 - x_2 \geq 100 \end{array}$$

→ each **unrestricted** variable (urs) is replaced by the **difference** of two new non-negative variables

$$\begin{array}{c}\ldots + x_1 + \ldots \\ x_1 \text{ urs}\end{array} \quad \Longleftrightarrow \quad \begin{array}{c}\ldots\ +\ (x_2 - x_3)\ +\ \ldots \\ x_2, x_3 \geq 0\end{array}$$

→ a **non-positive** variable $x_1 \leq 0$ is replaced by the **negative** of a new non-negative variable x_2

$$\begin{array}{c}\ldots + x_1 + \ldots \\ x_1 \leq 0\end{array} \quad \Longleftrightarrow \quad \begin{array}{c}\ldots\ +\ (-x_2)\ +\ \ldots \\ x_2 \geq 0\end{array}$$

→ **absolute** value: we can replace **only** in certain situations

- inequalities of type $|f| \leq g$ where f and g are arbitrary expressions:
  replace by two inequalities $f \leq g$ and $-g \leq f$
- if $+|f|$ appears in the **objective function** and we are **minimizing** this function:
  replace $+|f|$ in the objective function by a new variable x_1 and add a constraint $|f| \leq x_1$.
  (likewise if $-|f|$ appears when maximizing)

→ **min of max**: if $\max\{f_1, f_2, \ldots, f_t\}$ in the **objective function** and we are **minimizing**, then replace this expression with a new variable x_1 and add constraints $f_i \leq x_1$ for each $i = 1, \ldots, t$:

$$\ldots\ +\ \max\{f_1, \ldots, f_t\}\ +\ \ldots \quad \Longleftrightarrow \quad \begin{array}{c}\ldots\ +\ x_1\ + \ldots \\ x_1 \text{ urs}\end{array} \qquad \begin{array}{c} f_1 \leq x_1 \\ f_2 \leq x_1 \\ \vdots \\ f_t \leq x_1 \end{array}$$

→ **unrestricted expression** f can be written as a difference of two non-negative variables

$$\ldots + f + \ldots \quad \Longleftrightarrow \quad \begin{array}{c}\ldots\ +\ (x_2 - x_3)\ +\ \ldots \\ x_2, x_3 \geq 0\end{array}$$

Moreover, if we are **minimizing**, we can use $+x_2$ and $+x_3$ (**positive** multiples of x_2, x_3) in the **objective function** (if maximizing, we can use negative multiples).

In an **optimal solution** the meaning of these new variables will be as follows:

- if $f \geq 0$, then $x_2 = f$ and $x_3 = 0$,
- if $f < 0$, then $x_2 = 0$ and $x_3 = -f$.

In other words, x_2 represents the **positive part** of f, and x_3 the **negative part** of f (can you see why?). Note that this only guaranteed to hold for an optimal solution (but that will be enough for us).

Exercise. Try to justify for yourself why these restrictions are necessary.

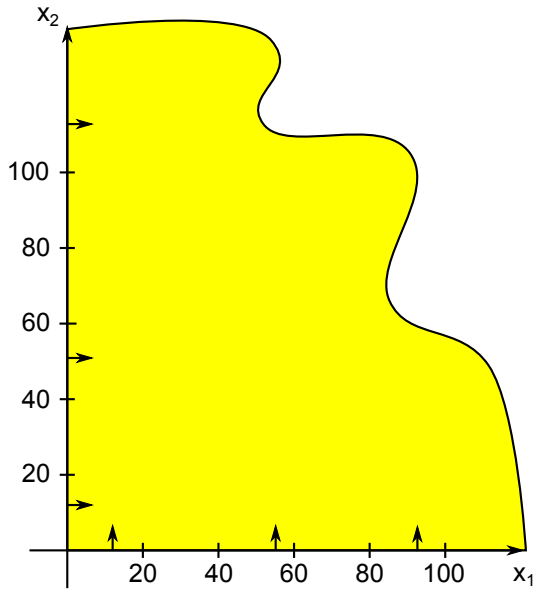
3 Solving linear programs

3.1 Graphical method

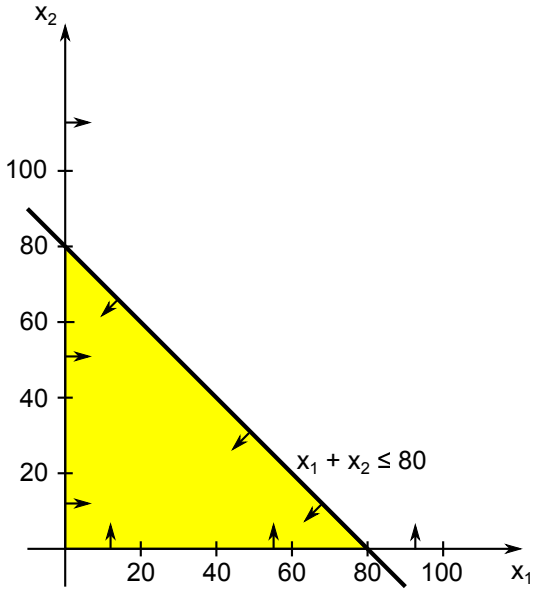
$$\begin{array}{rrcl} \text{Max} & 3x_1 + 2x_2 & & \\ & x_1 + x_2 & \leq & 80 \\ & 2x_1 + x_2 & \leq & 100 \\ & x_1 & \leq & 40 \\ & x_1, x_2 & \geq & 0 \end{array}$$

1. Find the feasible region.

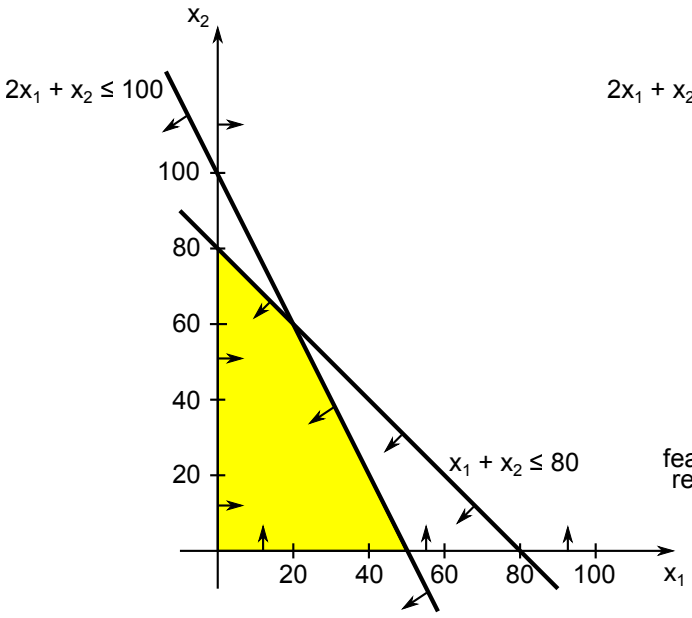
   - Plot each constraint as an equation $\equiv$ line in the plane
   - Feasible points on one side of the line – plug in (0,0) to find out which



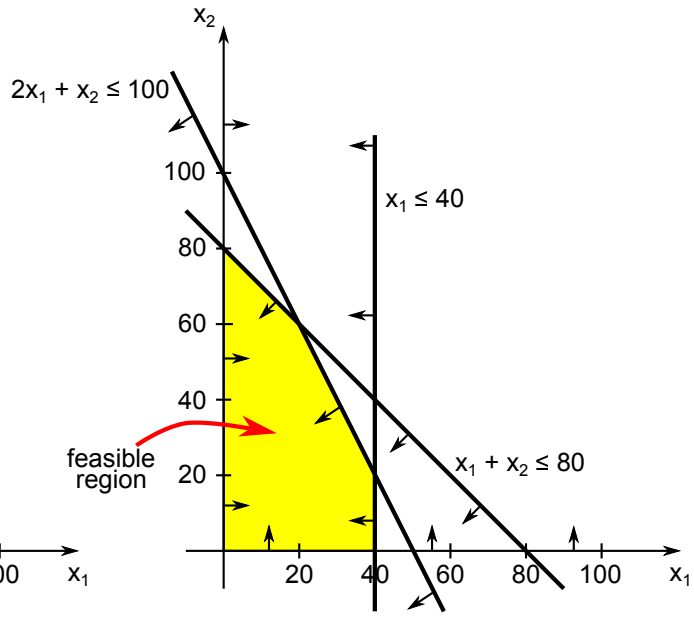
Start with $x_1 \geq 0$ and $x_2 \geq 0$



add $x_1 + x_2 \leq 80$



add $2x_1 + x_2 \leq 100$

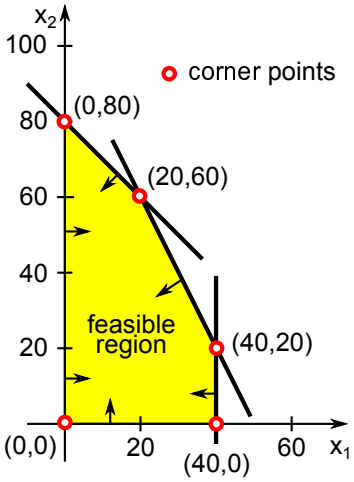


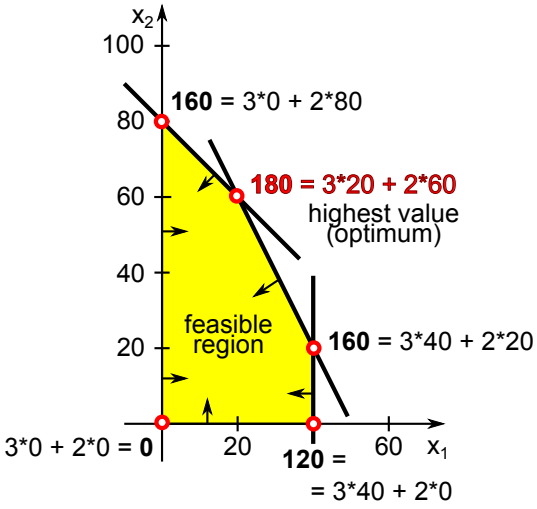
add $x_1 \leq 40$

A **corner** (extreme) point X of the region $R \equiv$ every line through X intersects R in a segment whose one endpoint is X. Solving a linear program amounts to finding a best corner point by the following theorem.

Theorem 1. *If a linear program has an* **optimal** *solution, then it also has an* **optimal solution** *that is a* **corner point** *of the feasible region.*

Exercise. Try to find all corner points. Evaluate the objective function $3x_1 + 2x_2$ at those points.





Problem: there may be too many corner points to check. There's a better way.

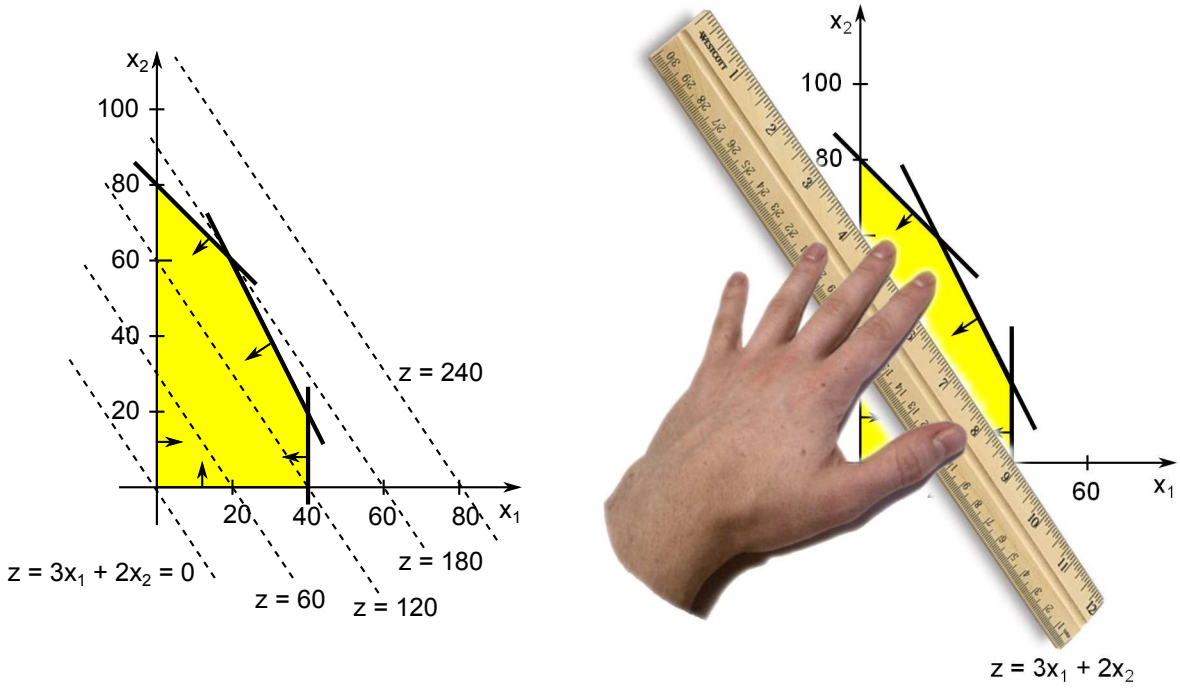
Iso-value line $\equiv$ in all points on this line the objective function has the same value

For our objective $3x_1 + 2x_2$ an iso-value line consists of points satisfying $3x_1 + 2x_2 = z$ where z is some number.

Graphical Method (main steps):

1. Find the feasible region.
2. Plot an iso-value (isoprofit, isocost) line for some value.

3. Slide the line in the direction of increasing value until it only touches the region.
4. Read-off an optimal solution.



Optimal solution is $(x_1, x_2) = (20, 60)$.

Observe that this point is the intersection of two lines forming the boundary of the feasible region. Recall that lines we use to construct the feasible region come from inequalities (the points on the line satisfy the particular inequality with equality).

Binding constraint $\equiv$ constraint satisfied with equality

For solution $(20, 60)$, the binding constraints are $x_1 + x_2 \leq 80$ and $2x_1 + x_2 \leq 100$ because $20 + 60 = 80$ and $2 \times 20 + 60 = 100$. The constraint $x_1 \leq 40$ is not binding because $x_1 = 20 < 40$.

The constraint is binding because changing it (a little) necessarily changes the optimality of the solution. Any change to the binding constraints either makes the solution not optimal or not feasible.

A constraint that is not binding can be changed (a little) without disturbing the optimality of the solution we found. Clearly we can change $x_1 \leq 40$ to $x_1 \leq 30$ and the solution $(20, 60)$ is still optimal. We shall discuss this more in-depth when we learn about Sensitivity Analysis.

Finally, note that the above process always yields one of the following cases.

Theorem 2. *Every linear program has either*

(i) a **unique** *optimal solution, or*
*(ii) multiple (***infinity***) optimal solutions, or*
(iii) is **infeasible** *(has no feasible solution), or*
(iv) is **unbounded** *(no feasible solution is maximal).*

3.2 Fourier-Motzkin Elimination (FME)

A simple (but not yet most efficient) process to solve linear programs. Unlike the Graphical method, this process applies to arbitrary linear programs, but more efficient methods exist. The FME method

finds a solution to a system of linear inequalities

(much like Gaussian elimination from Linear algebra which finds a solution to a system of **linear equations**)

We shall discuss how this is done for $\geq$-inequalities and for **minimization** LPs. (Similarly it can be stated for $\leq$-inequalities and maximization LPs.) You can skip to the example below to get a better idea.

First, we need to adapt the method to solving linear programs. We need to incorporate the objective function as part of the inequalities. We **replace** the objective function by a **new** variable z and look for a solution to the inequalities such that z is smallest possible (explained how later).

0. **Objective function** $c_1x_1 + c_2x_2 + \ldots + c_nx_n$: add a new constraint $z \geq c_1x_1 + c_2x_2 + \ldots + c_nx_n$

From this point, we assume that all we have is a system of $\geq$-inequalities with all variables on the left-hand side and a constant on the right-hand side. (We change $\leq$-inequalities to $\geq$-inequalities by multiplying by -1.) We proceed similarly as in Gaussian elimination. We try to eliminate variables one by one by **pivotting** a variable in **all inequalities** (not just one). Unlike Gaussian elimination, we are dealing with inequalities here and so we are **not allowed** to multiply by a negative constant when pivotting. This requires a more complex procedure to eliminate x_1.

1. **Normalize** x_1: if $+cx_1$ or $-cx_1$ where $c > 0$ appears in an inequality, divide the inequality by c.

   After normalizing, this gives us three types of inequalities: those with $+x_1$ (call them **positive** inequalities), those with $-x_1$ (call them **negative** inequalities), and those without x_1.

2. **Eliminate** x_1: consider each positive and each negative inequality and add them together to create a new inequality.

   Note that we do this for **every pair** of such inequalities; each generates a new inequality without x_1. Taking all these generated inequalities and the inequalities that did not contain x_1 in the first place gives us new problem, one without x_1. This new problem is **equivalent** to the original one.

3. **Repeat** this process eliminating $x_2, x_3, \ldots,$ in turn until only z remains to be eliminated.

4. **Solution:** determine the smallest value of z that satisfies the resulting inequalities.

5. **Back-substitution:** substitute the values in the reverse order of elimination to produce values of all eliminated variables.

   In this process, when choice is possible for some variable, we can choose arbitrarily; any choice leads to a correct solution (for more, see the example below how this is done).

Example

$$\begin{array}{lrcrcrcl} \min & 2x_1 & + & 2x_2 & + & 3x_3 & & \\ \text{s.t.} & x_1 & + & x_2 & + & x_3 & \leq & 2 \\ & 2x_1 & + & x_2 & & & \leq & 3 \\ & 2x_1 & + & x_2 & + & 3x_3 & \geq & 3 \\ & & & & & x_1, x_2, x_3 & \geq & 0 \end{array}$$

1. make the objective function into a constraint $z \geq \overbrace{2x_1 + 2x_2 + 3x_3}^{\text{objective function}}$ and change the objective to $\min z$

$$
\begin{array}{lrcrcrcrcl}
\min & & & & & & & z & & \\
\text{s.t.} & 2x_1 & + & 2x_2 & + & 3x_3 & - & z & \leq & 0 \\
 & x_1 & + & x_2 & + & x_3 & & & \leq & 2 \\
 & 2x_1 & + & x_2 & & & & & \leq & 3 \\
 & 2x_1 & + & x_2 & + & 3x_3 & & & \geq & 3 \\
 & & & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

2. change all inequalities to $\geq$

$$
\begin{array}{rcrcrcrcr}
-2x_1 & - & 2x_2 & - & 3x_3 & + & z & \geq & 0 \\
-x_1 & - & x_2 & - & x_3 & & & \geq & -2 \\
-2x_1 & - & x_2 & & & & & \geq & -3 \\
2x_1 & + & x_2 & + & 3x_3 & & & \geq & 3 \\
x_1 & & & & & & & \geq & 0 \\
 & & x_2 & & & & & \geq & 0 \\
 & & & & x_3 & & & \geq & 0
\end{array}
$$

3. Eliminate x_1

a) normalize = make the coefficients of x_1 one of $+1$, -1, or 0

$$
\begin{array}{rcrcrcrcr}
-x_1 & - & x_2 & - & \frac{3}{2}x_3 & + & \frac{1}{2}z & \geq & 0 \\
-x_1 & - & x_2 & - & x_3 & & & \geq & -2 \\
-x_1 & - & \frac{1}{2}x_2 & & & & & \geq & -\frac{3}{2} \\
x_1 & + & \frac{1}{2}x_2 & + & \frac{3}{2}x_3 & & & \geq & \frac{3}{2} \\
x_1 & & & & & & & \geq & 0 \\
 & & x_2 & & & & & \geq & 0 \\
 & & & & x_3 & & & \geq & 0
\end{array}
$$

b) add inequalities = add each inequality with $+x_1$ to every inequality with $-x_1$; then remove all inequalities containing x_1

$$
\begin{array}{rcrcrcrcr}
-\frac{1}{2}x_2 & & & + & \frac{1}{2}z & \geq & \frac{3}{2} \\
-\frac{1}{2}x_2 & + & \frac{1}{2}x_3 & & & \geq & -\frac{1}{2} \\
 & & \frac{3}{2}x_3 & & & \geq & 0 \\
 & - & \frac{3}{2}x_3 & + & \frac{1}{2}z & \geq & 0 \\
 & - & x_3 & & & \geq & -2 \\
-\frac{1}{2}x_2 & & & & & \geq & -\frac{3}{2} \\
x_2 & & & & & \geq & 0 \\
 & & x_3 & & & \geq & 0
\end{array}
$$

Eliminate x_2

$$
\begin{array}{rcrcrcr}
-x_2 & & & + & z & \geq & 3 \\
-x_2 & + & x_3 & & & \geq & -1 \\
-x_2 & - & \frac{3}{2}x_3 & + & \frac{1}{2}z & \geq & 0 \\
-x_2 & - & x_3 & & & \geq & -2 \\
-x_2 & & & & & \geq & -3 \\
x_2 & & & & & \geq & 0 \\
 & & \frac{3}{2}x_3 & & & \geq & 0 \\
 & & x_3 & & & \geq & 0
\end{array}
$$

$$
\begin{array}{rcrcr}
 & & z & \geq & 3 \\
 & x_3 & & \geq & -1 \\
 & -\frac{3}{2}x_3 & + & \frac{1}{2}z & \geq 0 \\
 & -x_3 & & \geq & -2 \\
 & & 0 & \geq & -3 \\
 & \frac{3}{2}x_3 & & \geq & 0 \\
 & x_3 & & \geq & 0
\end{array}
$$

Eliminate x_3

$$
\begin{array}{rcrcr}
-x_3 & + & \frac{1}{3}z & \geq & 0 \\
-x_3 & & & \geq & -2 \\
x_3 & & & \geq & -1 \\
x_3 & & & \geq & 0 \\
 & & z & \geq & 3 \\
 & & 0 & \geq & -3
\end{array}
$$

$$
\begin{array}{rcr}
\frac{1}{3}z & \geq & -1 \\
0 & \geq & -3 \\
\frac{1}{3}z & \geq & 0 \\
0 & \geq & -2 \\
z & \geq & 3 \\
0 & \geq & -3
\end{array}
$$

Final list of inequalities

$$\begin{aligned} z &\geq -3 \\ z &\geq 0 \\ z &\geq 3 \\ 0 &\geq -3 \\ 0 &\geq -2 \end{aligned}$$

4. Choose smallest z that satisfies the inequalities, $z = 3$

5. Back-substitution

$$\begin{aligned} -x_3 + \tfrac{1}{3} \times 3 &\geq 0 \\ -x_3 &\geq -2 \\ x_3 &\geq -1 \\ x_3 &\geq 0 \end{aligned} \qquad \begin{aligned} x_3 &\leq 1 \\ x_3 &\leq 2 \\ x_3 &\geq -1 \\ x_3 &\geq 0 \end{aligned}$$

$$0 \leq x_3 \leq 1$$

Choose ANY value that satisfies the inequalities

$$\boxed{x_3 = \tfrac{1}{2}}$$

$$\begin{aligned} -x_2 + 3 &\geq 3 \\ -x_2 + \tfrac{1}{2} &\geq -1 \\ -x_2 - \tfrac{3}{2} \times \tfrac{1}{2} + \tfrac{1}{2} \times 3 &\geq 0 \\ -x_2 - \tfrac{1}{2} &\geq -2 \\ -x_2 &\geq -3 \\ x_2 &\geq 0 \end{aligned} \qquad \begin{aligned} x_2 &\leq 0 \\ x_2 &\leq \tfrac{3}{2} \\ x_2 &\leq \tfrac{3}{4} \\ x_2 &\leq \tfrac{3}{2} \\ x_2 &\leq 3 \\ x_2 &\geq 0 \end{aligned}$$

$$0 \leq x_2 \leq 0$$

Choose ANY value that satisfies the inequalities (this time only one)

$$\boxed{x_2 = 0}$$

$$\begin{aligned} -x_1 - 0 - \tfrac{3}{2} \times \tfrac{1}{2} + \tfrac{1}{2} \times 3 &\geq 0 \\ -x_1 - 0 - \tfrac{1}{2} &\geq -2 \\ -x_1 - \tfrac{1}{2} \times 0 &\geq -\tfrac{3}{2} \\ x_1 + \tfrac{1}{2} \times 0 + \tfrac{3}{2} \times \tfrac{1}{2} &\geq \tfrac{3}{2} \\ x_1 &\geq 0 \end{aligned} \qquad \begin{aligned} x_1 &\leq \tfrac{3}{4} \\ x_1 &\leq \tfrac{3}{2} \\ x_1 &\leq \tfrac{3}{2} \\ x_1 &\geq \tfrac{3}{4} \\ x_1 &\geq 0 \end{aligned}$$

$$\tfrac{3}{4} \leq x_1 \leq \tfrac{3}{4}$$

Choose ANY value that satisfies the inequalities (again only one)

$$\boxed{x_1 = \tfrac{3}{4}}$$

$$\boxed{\text{Solution } x_1 = \tfrac{3}{4},\ x_2 = 0,\ x_3 = \tfrac{1}{2} \text{ of value } z = 3}$$

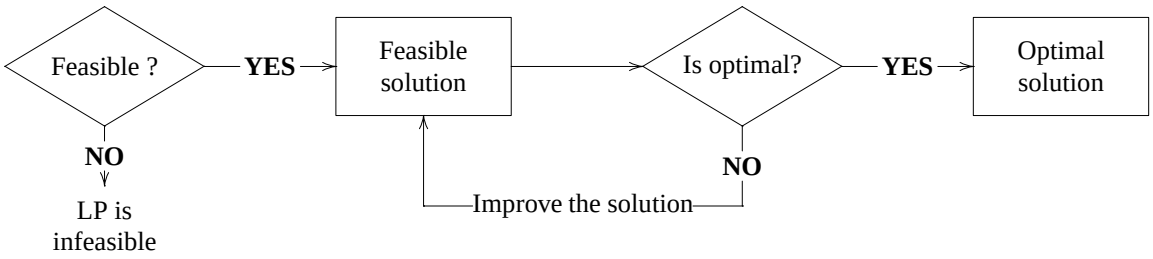
Notes:

- if at any point an inequality $0x_1 + 0x_2 + 0x_3 + 0z \geq d$ is produced where $d > 0$
  $\longrightarrow$ no solution (**infeasible LP**)
- if the final system does not contain an inequality $z \geq d$
  $\longrightarrow$ no optimum (**unbounded LP**)

4 Simplex method

The process consists of two steps

1. Find a **feasible** solution (or determine that **none exists**).
2. Improve the feasible solution to an **optimal** solution.



In many cases the first step is easy (for free; more on that later).

4.1 Canonical form

Linear program (LP) is in a **canonical form** if

- all constraints are **equations**
- all variables are **non-negative**

$$
\begin{array}{lccccccccc}
\max & c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n & & \\
\text{subject to} & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\
& \vdots & + & \vdots & & & & \vdots & & \vdots \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m \\
& & & & & & & x_1, x_2, \dots, x_n & \geq & 0
\end{array}
$$

Slack variables

To change a inequality to an equation, we add a **new non-negative** variable called a **slack** variable.

$$x_1 + x_2 \leq 80 \qquad \longrightarrow \qquad x_1 + x_2 + s_1 = 80$$

$$x_1 + x_2 \geq 80 \qquad \longrightarrow \qquad x_1 + x_2 - e_1 = 80$$

Notes:

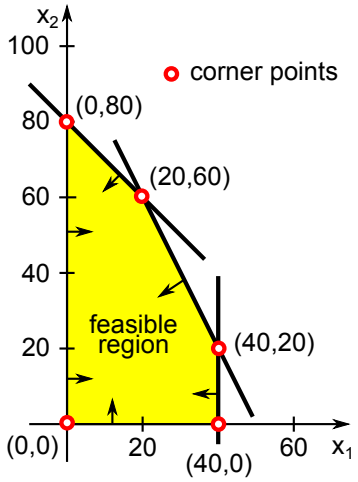
- the variable e_1 is sometimes called an **excess** variable
- we can use x_i for slack variables (where i is a new index)

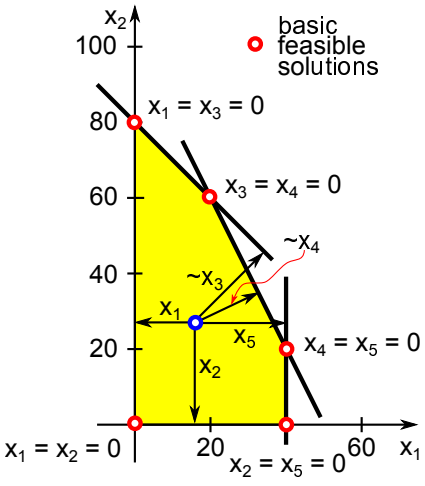
$$\begin{array}{rrrrl} \max & 3x_1 & + & 2x_2 & \\ & x_1 & + & x_2 & \leq 80 \\ & 2x_1 & + & x_2 & \leq 100 \\ & x_1 & & & \leq 40 \\ & & x_1, x_2 & & \geq 0 \end{array} \quad \longrightarrow \quad \begin{array}{rrrrrrrrl} \max & 3x_1 & + & 2x_2 & & & & & \\ & x_1 & + & x_2 & + & x_3 & & & = 80 \\ & 2x_1 & + & x_2 & & + & x_4 & & = 100 \\ & x_1 & & & & & & + x_5 & = 40 \\ & & & & x_1, x_2, & x_3, & x_4, & x_5 & \geq 0 \end{array}$$

4.2 Simplex method by example

Consider the toyshop example from earlier lectures. Convert to equalities by adding *slack variables*

$$\begin{array}{rrrrl} \max & 3x_1 & + & 2x_2 & \\ & x_1 & + & x_2 & \leq 80 \\ & 2x_1 & + & x_2 & \leq 100 \\ & x_1 & & & \leq 40 \\ & & x_1, x_2 & & \geq 0 \end{array} \quad \longrightarrow \quad \begin{array}{rrrrrrrrl} \max & 3x_1 & + & 2x_2 & & & & & \\ & x_1 & + & x_2 & + & x_3 & & & = 80 \\ & 2x_1 & + & x_2 & & + & x_4 & & = 100 \\ & x_1 & & & & & & + x_5 & = 40 \\ & & & & x_1, x_2, & x_3, & x_4, & x_5 & \geq 0 \end{array}$$





Starting feasible solution

Set variables x_1, x_2 to zero and set slack variables to the values on the right-hand side.

$\rightarrow$ yields a feasible solution $x_1 = x_2 = 0$, $x_3 = 80$, $x_4 = 100$, $x_5 = 40$

Recall that the solution is feasible because all variables are **non-negative** and **satisfy** all equations.

(we get a feasible solution right away because the right-hand side is non-negative; this may not always work)

Note something **interesting**: in this feasible solution two variables (namely x_1, x_2) are zero. Such a solution is called a **basic solution** of this problem, because the value of at least two variables is zero.

In a problem with n variables and m constraints, a solution where at least $(n - m)$ variables are zero is a **basic solution**.

A basic solution that is also feasible is called a **basic feasible solution** (BFS).

The importance of basic solutions is revealed by the following observation.

Basic solutions are precisely the **corner points** of the **feasible region**.

Recall that we have discussed that to find an optimal solution to an LP, it suffices to find a **best solution** among all **corner points**. The above tells us how to compute them – they are the **basic feasible solutions**.

A variable in a **basic solution** is called a **non-basic variable** if it is chosen to be zero.
Otherwise, the variable is **basic**.

The basic variables we collectively call a **basis**.

Dictionary

To conveniently deal with basic solutions, we use the so-called **dictionary**. A dictionary lists values of basic variables as a function of non-basic variables. The correspondence is obtained by expressing the basic variables from the initial set of equations. (We shall come back to this later; for now, have a look below.)

Express the slack variables from the individual equations.

$$\begin{array}{rrrrrrrrl} \max & 3x_1 & + & 2x_2 & & & & & \\ & x_1 & + & x_2 & + & x_3 & & = & 80 \\ & 2x_1 & + & x_2 & & + & x_4 & = & 100 \\ & x_1 & & & & & + & x_5 = & 40 \\ & & & & & x_1, x_2, x_3, x_4, x_5 & \geq & & 0 \end{array} \quad \longrightarrow \quad \begin{array}{rcl} x_3 &=& 80 - x_1 - x_2 \\ x_4 &=& 100 - 2x_1 - x_2 \\ x_5 &=& 40 - x_1 \\ \hline z &=& 0 + 3x_1 + 2x_2 \end{array}$$

This is called a **dictionary** .

- x_1, x_2 independent (**non-basic**) variables
- x_3, x_4, x_5 dependent (**basic**) variables
- $\{x_3, x_4, x_5\}$ is a **basis**

set $x_1 = x_2 = 0 \rightarrow$ the corresponding (feasible) solution is $x_3 = 80$, $x_4 = 100$, $x_5 = 40$ with value $z = 0$

Improving the solution

Try to increase x_1 from its current value 0 in hopes of improving the value of z

try $x_1 = 20$, $x_2 = 0$ and **substitute** into the dictionary to obtain the values of x_3, x_4, x_5 and z
$\longrightarrow x_3 = 60$, $x_4 = 60$, $x_5 = 20$ with value $z = 60 \rightarrow$ **feasible**

try again $x_1 = 40$, $x_2 = 0 \longrightarrow x_3 = 40$, $x_4 = 20$, $x_5 = 0$ with value $z = 120 \rightarrow$ **feasible**

now try $x_1 = 50$, $x_2 = 0 \longrightarrow x_3 = 30$, $x_4 = 0$, $x_5 = -10 \rightarrow$ **not feasible**

How much we can increase x_1 before a (dependent) variable becomes negative?

If $x_1 = t$ and $x_2 = 0$, then the solution is feasible if

$$\left.\begin{array}{rcl} x_3 = 80 - t - 0 \geq 0 \\ x_4 = 100 - 2t - 0 \geq 0 \\ x_5 = 40 - t \geq 0 \end{array}\right. \quad \Longleftrightarrow \quad \left.\begin{array}{l} t \leq 80 \\ t \leq 50 \\ t \leq 40 \end{array}\right\} \Longleftrightarrow \quad t \leq 40$$

Maximal value is $x_1 = 40$ at which point the variable x_5 becomes zero

x_1 is *incoming* variable and x_5 is *outgoing* variable

(we say that x_1 **enters** the dictionary/basis, and x_5 **leaves** the dictionary/basis)

Ratio test

The above analysis can be streamlined into the following simple "ratio" test.

$x_3 : \frac{80}{1} = 80$

$x_4 : \frac{100}{2} = 50$

$x_5 : \frac{40}{1} = 40$

$$\begin{array}{rcrcrcr} x_3 & = & 80 & - & x_1 & - & x_2 \\ x_4 & = & 100 & - & 2x_1 & - & x_2 \\ x_5 & = & 40 & - & x_1 & & \\ \hline z & = & 0 & + & 3x_1 & + & 2x_2 \end{array}$$

ratio for x_4: $\frac{100}{2} = 50$

(**watch-out:** we only consider this ratio because the coefficient of x_1 is negative (-2)... more on that in the later steps)

Minimum achieved with $x_5 \Longrightarrow$ outgoing variable

Express x_1 from the equation for x_5

$$x_5 = 40 - x_1 \quad \longrightarrow \quad x_1 = 40 - x_5$$

Substitute x_1 to all other equations $\longrightarrow$ new feasible dictionary

$$\begin{array}{rcl} x_1 & = & (40 - x_5) \\ x_3 & = & 80 - (40 - x_5) - x_2 \\ x_4 & = & 100 - 2(40 - x_5) - x_2 \\ \hline z & = & 0 + 3(40 - x_5) + 2x_2 \end{array} \quad \longrightarrow \quad \begin{array}{rcl} x_1 & = & 40 - x_5 \\ x_3 & = & 40 - x_2 + x_5 \\ x_4 & = & 20 - x_2 + 2x_5 \\ \hline z & = & 120 + 2x_2 - 3x_5 \end{array}$$

now x_2, x_5 are independent variables and x_1, x_3, x_4 are dependent

$\rightarrow \{x_1, x_3, x_4\}$ is a basis

we repeat: we increase $x_2 \rightarrow$ incoming variable, ratio test:

x_1 : does not contain $x_2 \rightarrow$ no constraint

$x_2 : \frac{40}{1} = 40$

$x_4 : \frac{20}{1} = 20$

minimum achieved for $x_4 \rightarrow$ outgoing variable

$$x_4 = 20 - x_2 + 2x_5 \quad \longrightarrow \quad x_2 = 20 - x_4 + 2x_5$$

$$\begin{array}{rcl} x_1 & = & 40 - x_5 \\ x_2 & = & (20 - x_4 + 2x_5) \\ x_3 & = & 40 - (20 - x_4 + 2x_5) + x_5 \\ \hline z & = & 120 + 2(20 - x_4 + 2x_5) - 3x_5 \end{array} \quad \longrightarrow \quad \begin{array}{rcl} x_1 & = & 40 - x_5 \\ x_2 & = & 20 - x_4 + 2x_5 \\ x_3 & = & 20 + x_4 - x_5 \\ \hline z & = & 160 - 2x_4 + x_5 \end{array}$$

x_5 incoming variable, ratio test:

$x_1 : \frac{40}{1} = 40$

x_2 : positive coefficient $\rightarrow$ no constraint

$x_3 : \frac{20}{1} = 20$

minimum achieved for $x_3 \rightarrow$ outgoing variable

$$x_3 = 20 + x_4 - x_5 \quad \longrightarrow \quad x_5 = 20 + x_4 - x_3$$

$$\begin{array}{rcl} x_1 & = & 40 - (20 + x_4 - x_3) \\ x_2 & = & 20 - x_4 + 2(20 + x_4 - x_3) \\ x_5 & = & (20 + x_4 - x_3) \\ \hline z & = & 160 - 2x_4 + (20 + x_4 - x_3) \end{array} \quad \longrightarrow \quad \begin{array}{rcl} x_1 & = & 20 + x_3 - x_4 \\ x_2 & = & 60 - 2x_3 + x_4 \\ x_5 & = & 20 - x_3 + x_4 \\ \hline z & = & 180 - x_3 - x_4 \end{array}$$

no more improvement possible $\longrightarrow$ **optimal solution**

$$x_1 = 20, x_2 = 60, x_3 = 0, x_4 = 0, x_5 = 20 \text{ of value } z = 180$$

Why? setting x_3, x_4 to any non-zero values results in a smaller value of z

Each dictionary is **equivalent** to the original system (the two have the same set of solutions)

Simplex algorithm

Preparation: find a starting feasible solution/dictionary

1. Convert to the canonical form (constraints are equalities) by adding slack variables $x_{n+1}, \ldots, x_{n+m}$
2. Construct a starting dictionary - express slack variables and objective function z
3. If the resulting dictionary is feasible, then we are done with preparation
   If not, try to find a feasible dictionary using the **Phase I. method** (next lecture).

Simplex step (maximization LP): try to improve the solution

1. **(Optimality test)**: If **no variable** appears with a **positive** coefficient in the equation for z
   $\rightarrow$ STOP, current solution is **optimal**
   - set non-basic variables to zero
   - read off the values of the basic variables and the objective function z
     $\rightarrow$ Hint: the values are the constant terms in respective equations
   - report this (optimal) solution
2. Else pick a variable x_i having positive coefficient in the equation for z
   $x_i \equiv$ *incoming* variable
3. Ratio test: in the dictionary, find an equation for a variable x_j in which
   - x_i appears with a negative coefficient $-a$
   - the ratio $\frac{b}{a}$ is smallest possible
     (where b is the constant term in the equation for x_j)
4. If no such such x_j exists $\rightarrow$ stop, no optimal solution, report that **LP is unbounded**
5. Else $x_j \equiv$ outgoing variable $\rightarrow$ construct a new dictionary by *pivoting*:
   - express x_i from the equation for x_j,
   - add this as a new equation,
   - remove the equation for x_j,
   - substitute x_i to all other equations (including the one for z)
6. Repeat from 1.

Questions:

- which variable to choose as incoming, which as outgoing
- is this guaranteed to terminate in a finite number of steps
- how to convert other LP formulations to the standard form
- how to find a starting dictionary
- how do we find alternative optimal solutions

4.3 Two phase Simplex method

canonical form = equations, non-negative variables

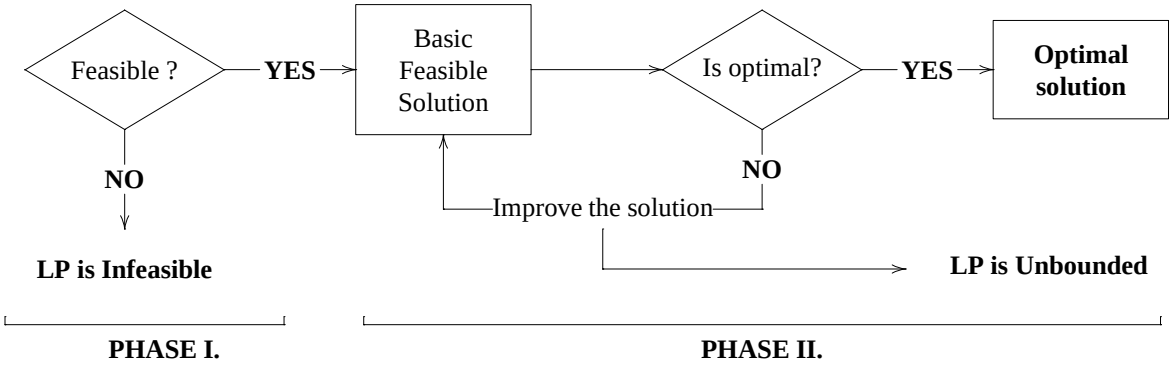
n = number of variables

m = number of equations

basic solution = at least $(n-m)$ variables are zero

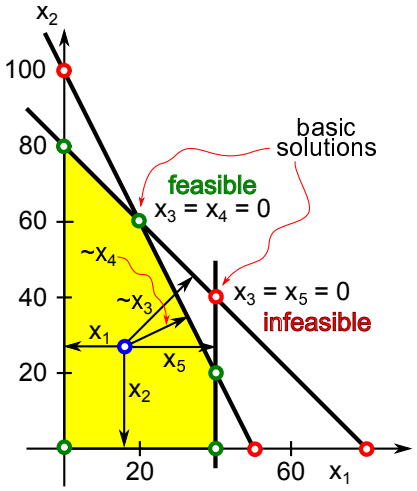
basic solutions = dictionaries

basic feasible solutions = corner/extreme points = feasible dictionaries



$$\begin{aligned} \max\ & 3x_1 + 2x_2 \\ & x_1 + x_2 \le 80 \\ & 2x_1 + x_2 \le 100 \\ & x_1 \le 40 \\ & x_1, x_2 \ge 0 \end{aligned}$$

$$\begin{aligned} \max\ & 3x_1 + 2x_2 \\ & x_1 + x_2 + x_3 = 80 \\ & 2x_1 + x_2 + x_4 = 100 \\ & x_1 + x_5 = 40 \\ & x_1, x_2, x_3, x_4, x_5 \ge 0 \end{aligned}$$



Feasible dictionary: $x_3 = x_4 = 0$

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned}$$

Infeasible dictionary: $x_3 = x_5 = 0$

$$\begin{aligned} x_1 &= 40 - x_5 \\ x_2 &= 40 - x_3 + x_5 \\ x_4 &= -20 + x_3 + x_5 \\ \hline z &= 200 - 2x_3 - x_5 \end{aligned}$$

(it is infeasible since $x_4 = -20$)

4.4 Special cases

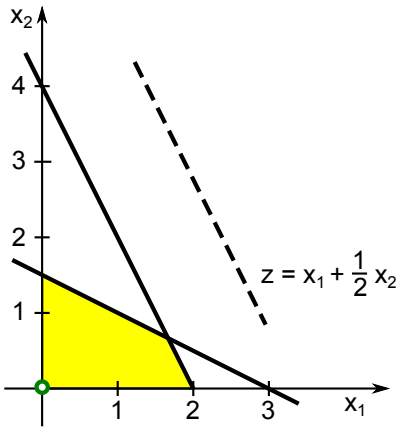
Alternative solutions

$$\begin{aligned} \max\ & x_1 + \tfrac{1}{2}x_2 \\ \text{s.t. } & 2x_1 + x_2 \le 4 \\ & x_1 + 2x_2 \le 3 \\ & x_1, x_2 \ge 0 \end{aligned}$$

$$\begin{aligned} x_3 &= 4 - 2x_1 - x_2 \\ x_4 &= 3 - x_1 - 2x_2 \\ \hline z &= x_1 + \tfrac{1}{2}x_2 \end{aligned}$$

Pivoting: x_1 enters, ratio test: $x_3 : \dfrac{4}{2} = 2$, $x_4 : \dfrac{3}{1} = 3 \longrightarrow$ thus x_3 leaves: $x_1 = 2 - \frac{1}{2}x_2 - \frac{1}{2}x_3$

$$\begin{aligned} x_1 &= 2 - \tfrac{1}{2}x_2 - \tfrac{1}{2}x_3 \\ x_4 &= 1 - \tfrac{3}{2}x_2 + \tfrac{1}{2}x_3 \\ \hline z &= 2 + 0x_2 - \tfrac{1}{2}x_3 \end{aligned}$$



Optimal solution (all coefficients non-positive) $x_1 = 2$, $x_2 = 0$, $x_3 = 0$, $x_4 = 1$, $z = 2$

Note that x_2 appears with zero coeeficient in the expression for z

$\rightarrow$ increasing x_2 is possible, but does not affect the value of z

we pivot again: x_2 enters, ratio test $x_1 : \dfrac{2}{1/2} = 4$, $x_4 : \dfrac{1}{3/2} = 2/3 \longrightarrow$ thus x_4 leaves

$$\begin{aligned} x_1 &= \tfrac{5}{3} - \tfrac{2}{3}x_3 + \tfrac{1}{3}x_4 \\ x_2 &= \tfrac{2}{3} + \tfrac{1}{3}x_3 - \tfrac{2}{3}x_4 \\ \hline z &= 2 - \tfrac{1}{2}x_3 + 0x_4 \end{aligned}$$

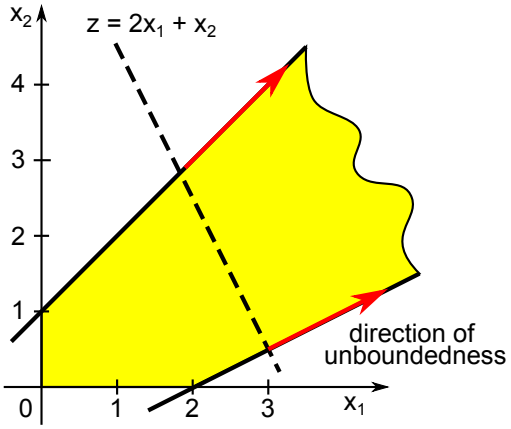
Again an optimal solution $x_1 = \frac{5}{3}$, $x_2 = \frac{2}{3}$, $x_3 = 0$, $x_4 = 0$, $z = 2 \rightarrow$ same value

What if we pivot again (on x_4) ?

Unbounded LP

$$\begin{aligned} \max\ & 2x_1 + x_2 \\ \text{s.t. } & -x_1 + x_2 \le 1 \\ & x_1 - 2x_2 \le 2 \\ & x_1, x_2 \ge 0 \end{aligned}$$

$$\begin{aligned} x_3 &= 1 + x_1 - x_2 \\ x_4 &= 2 - x_1 + 2x_2 \\ \hline z &= 2x_1 + x_2 \end{aligned}$$



Pivoting: x_1 enters, x_4 leaves (the only choice), $x_1 = 2 + 2x_2 - x_4$

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

for $x_2 = x_4 = 0$, we have $x_1 = 2$, $x_3 = 3$, $z = 4$

$\longrightarrow$ a feasible solution

What if we now increase x_2? no positive value of x_2 makes one of x_1, x_3 negative

$\rightarrow$ we can make x_2 arbitrarily large and thus make z arbitrarily large $\longrightarrow$ **unbounded** LP

direction of unboundedness: set $x_2 = t$, $x_4 = 0 \longrightarrow x_1 = 2 + 2t$, $x_3 = 3 + t$, $z = 4 + 5t$

for increasing $t \longrightarrow$ gives a sequence of feasible solution of increasing value

Degeneracy

$$\begin{array}{rcl} x_4 &=& 1 \qquad\qquad - 2x_3 \\ x_5 &=& 3 - 2x_1 + 4x_2 - 6x_3 \\ x_6 &=& 2 + x_1 - 3x_2 - 4x_3 \\ \hline z &=& 2x_1 - x_2 + 8x_3 \end{array}$$

Pivotting: x_3 enters, ratio test: $x_4 : \frac{1}{2} = 1/2$, $x_5 : \frac{3}{6} = 1/2$, $x_6 : \frac{2}{4} = 1/2 \longrightarrow$ any of x_4, x_5, x_6 can be chosen

$\rightarrow$ we choose x_4 to leave, $x_3 = \frac{1}{2} - \frac{1}{2}x_4$

$$\begin{array}{rcl} x_3 &=& \frac{1}{2} \qquad\qquad - \frac{1}{2}x_4 \\ x_5 &=& - 2x_1 + 4x_2 + 3x_4 \\ x_6 &=& x_1 - 3x_2 + 2x_4 \\ \hline z &=& 4 + 2x_1 - x_2 - 4x_4 \end{array}$$

setting $x_1 = x_2 = x_4 = 0$ yields $x_3 = \frac{1}{2}$, $x_5 = 0$, $x_6 = 0$

now x_1 enters, and x_5 leaves (the only choice), $x_1 = 2x_2 - \frac{3}{2}x_4 - \frac{1}{2}x_5$

$$\begin{array}{rcl} x_1 &=& 2x_2 + \frac{3}{2}x_4 - \frac{1}{2}x_5 \\ x_3 &=& \frac{1}{2} \qquad - \frac{1}{2}x_4 \\ x_6 &=& - x_2 + \frac{7}{2}x_4 - \frac{1}{2}x_5 \\ \hline z &=& 4 + 3x_2 - x_4 - x_5 \end{array}$$

setting $x_2 = x_4 = x_5 = 0$ yields $x_1 = 0$, $x_3 = \frac{1}{2}$, $x_6 = 0$

$\longrightarrow$ **same solution** as before

if some basic variable is zero, then the basic solution is **degenerate**

This happens, for instance, if there is more than one choice for an outgoing variable (the ones not chosen will be zero in the subsequent dictionary)

Problem: several dictionaries may correspond to the same (degenerate) solution

The simplex rule may cycle, it is possible to go back to the same dictionary if we are not careful enough when choosing the incoming/outgoing variables

Bland's rule From possible options, choose an incoming (outgoing) variable x_k with smallest subscript k.

Simplex method using Bland's rule is guaranteed to terminate in a finite number of steps.

Alternative: **lexicographic rule** – choose as outgoing variable one whose row is lexicographically smallest (when divided by the constant term) – the coefficients in the objective function are guaranteed to strictly increase lexicographically

4.5 Phase I.

$$\begin{array}{lrcl} \max & x_1 - x_2 + x_3 & & \\ \text{s.t.} & 2x_1 - x_2 + 2x_3 &\leq& 4 \\ & 2x_1 - 3x_2 + x_3 &\leq& -5 \\ & -x_1 + x_2 - 2x_3 &\leq& -1 \\ & x_1, x_2, x_3 &\geq& 0 \end{array}$$

$$\begin{array}{rcl} x_4 &=& 4 - 2x_1 + x_2 - 2x_3 \\ x_5 &=& -5 - 2x_1 + 3x_2 - x_3 \\ x_6 &=& -1 + x_1 - x_2 + 2x_3 \\ \hline z &=& x_1 - x_2 + x_3 \end{array}$$

If we choose the starting basis to be the **slack variables**, then the resulting dictionary is **not feasible**:

$\rightarrow$ we let $x_1 = x_2 = x_3 = 0$, we get $x_4 = 4$, $x_5 = -5$, $x_6 = -1 \longrightarrow$ **not feasible** because $x_5 < 0$

We need to find a starting feasible dictionary for Phase II. To do this, we **solve** a different problem.

Intuition: we want to get a starting feasible solution where all variables x_1, x_2, x_3 are zero.

Option 1

Add new **artificial** variables to each inequality as follows:

$$\begin{aligned} x_1 + x_2 &\leq -100 \quad \longrightarrow \quad x_1 + x_2 - a_1 \leq -100 \\ x_1 + x_2 &\geq 100 \quad \longrightarrow \quad x_1 + x_2 + a_1 \geq 100 \\ x_1 + x_2 &= 100 \quad \longrightarrow \quad x_1 + x_2 + a_1 = 100 \\ x_1 + x_2 &= -100 \quad \longrightarrow \quad x_1 + x_2 - a_1 = -100 \end{aligned}$$

$$\left.\begin{aligned} x_1 + x_2 &\leq 100 \\ x_1 + x_2 &\geq -100 \\ x_1 + x_2 &= 0 \end{aligned}\right\} \longrightarrow \text{no change}$$

New objective function: minimize the **sum** of **all artificial** variables $a_1 + a_2 + \ldots + a_m$

Observe that setting all variables x_i to be 0 allows us to choose non-negative values for the artificial variables ($a_1 = 100$ in the above) to obtain a **starting feasible solution** for this **new problem**.

$$\begin{array}{lrcl} \max & x_1 - x_2 + x_3 & & \\ \text{s.t.} & 2x_1 - x_2 + 2x_3 & \leq & 4 \\ & 2x_1 - 3x_2 + x_3 & \leq & -5 \\ & -x_1 + x_2 - 2x_3 & \leq & -1 \\ & x_1, x_2, x_3 & \geq & 0 \end{array} \qquad \begin{array}{lrcl} \min & a_2 + a_3 & & \\ \text{s.t.} & 2x_1 - x_2 + 2x_3 & \leq & 4 \\ & 2x_1 - 3x_2 + x_3 - a_2 & \leq & -5 \\ & -x_1 + x_2 - 2x_3 - a_3 & \leq & -1 \\ & x_1, x_2, x_3, a_2, a_3 & \geq & 0 \end{array}$$

Notice that if we now set $x_1 = x_2 = x_3 = 0$ and $a_2 = 5$ and $a_3 = 1$, this satisfies all inequalities (is **feasible**). After adding the slack variables, we produce the corresponding **feasible dictionary** as follows. Since we want the **maximization form**, we also re write the objective function as $\max w = -a_2 - a_3$. Therefore

$$\begin{array}{lrcl} \max w = & -a_2 - a_3 & & \\ \text{s.t.} & 2x_1 - x_2 + 2x_3 + x_4 & = & 4 \\ & 2x_1 - 3x_2 + x_3 - a_2 + x_5 & = & -5 \\ & -x_1 + x_2 - 2x_3 - a_3 + x_6 & = & -1 \\ & x_1, x_2, x_3, a_2, a_3, x_4, x_5, x_6 & \geq & 0 \end{array}$$

If the optimal solution to this problem has **negative** value w, then the initial LP is **Infeasible**.
Otherwise, we produce a **starting** feasible dictionary for Phase II from the optimal dictionary of Phase I.

To get a starting (Phase I.) feasible dictionary, we take as a basis **all artificial** variables (a_2, a_3) and add to that **slack variables** of equations that do not have artificial variables (1st equation, add x_4).

$$\begin{aligned} x_4 &= 4 - 2x_1 + x_2 - 2x_3 \\ a_2 &= 5 + 2x_1 - 3x_2 + x_3 + x_5 \\ a_3 &= 1 - x_1 + x_2 - 2x_3 + x_6 \end{aligned}$$

The final step is the objective function $w = -a_2 - a_3$ which we have to write in terms of **non-basic** variables (so far it is not since we chose a_2 and a_3 to be basic). We substitute from the above equations:

$$w = -a_2 - a_3 = -\overbrace{(5 + 2x_1 - 3x_2 + x_3 + x_5)}^{a_2} - \overbrace{(1 - x_1 + x_2 - 2x_3 + x_6)}^{a_3} = -6 - x_1 + 2x_2 + x_3 - x_5 - x_6$$

The resulting starting feasible dictionary it then as follows:

$$\begin{array}{rcl} x_4 = & 4 - 2x_1 + x_2 - 2x_3 & \\ a_2 = & 5 + 2x_1 - 3x_2 + x_3 + x_5 & \\ a_3 = & 1 - x_1 + x_2 - 2x_3 + x_6 & \\ \hline w = & -6 - x_1 + 2x_2 + x_3 - x_5 - x_6 & \end{array}$$

Option 2

(for $\leq$ inequalities): Introduce **one** new artificial variable x_0 and a new objective $w = -x_0$

$$\begin{array}{lrcl} \max & -x_0 & & \\ \text{s.t.} & 2x_1 - x_2 + 2x_3 - x_0 & \leq & 4 \\ & 2x_1 - 3x_2 + x_3 - x_0 & \leq & -5 \\ & -x_1 + x_2 - 2x_3 - x_0 & \leq & -1 \\ & x_0, x_1, x_2, x_3 & \geq & 0 \end{array} \qquad \begin{array}{lrcl} \max & -x_0 & & \\ \text{s.t.} & 2x_1 - x_2 + 2x_3 - x_0 + x_4 & = & 4 \\ & 2x_1 - 3x_2 + x_3 - x_0 + x_5 & = & -5 \\ & -x_1 + x_2 - 2x_3 - x_0 + x_6 & = & -1 \\ & x_0, x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0 \end{array}$$

It is easy to get a starting feasible solution for this problem – a starting feasible basis is as follows:

- take all slack variables (x_4, x_5, x_6)
- consider the inequality whose right-hand side is most negative (in this case 2nd inequality)
- this inequality has an associated slack variable (x_5), remove this variable from our set $\rightarrow \{x_4, x_6\}$
- add x_0 in place of the removed variable $\rightarrow \{x_0, x_4, x_6\}$

This is guaranteed to be a feasible basis (in this new problem).

$$\begin{array}{rcl} x_0 &=& 5 + 2x_1 - 3x_2 + x_3 + x_5 \\ x_4 &=& 9 - 2x_2 - x_3 + x_5 \\ x_6 &=& 4 + 3x_1 - 4x_2 + 3x_3 + x_5 \\ \hline w &=& -5 - 2x_1 + 3x_2 - x_3 - x_5 \end{array}$$

Example

Let us solve the above problem.

x_2 enters, ratio test: $x_0 : \frac{5}{3}$, $x_4 : \frac{9}{2}$, $x_6 : \frac{4}{4} = 1 \rightarrow$ thus x_6 leaves, $x_2 = 1 + \frac{3}{4}x_1 + \frac{3}{4}x_3 + \frac{1}{4}x_5 - \frac{1}{4}x_6$

$$\begin{array}{rcl} x_0 &=& 2 - \frac{1}{4}x_1 - \frac{5}{4}x_3 + \frac{1}{4}x_5 + \frac{3}{4}x_6 \\ x_2 &=& 1 + \frac{3}{4}x_1 + \frac{3}{4}x_3 + \frac{1}{4}x_5 - \frac{1}{4}x_6 \\ x_4 &=& 7 - \frac{3}{2}x_1 - \frac{5}{2}x_3 + \frac{1}{2}x_5 + \frac{1}{2}x_6 \\ \hline w &=& -2 + \frac{1}{4}x_1 + \frac{5}{4}x_3 - \frac{1}{4}x_5 - \frac{3}{4}x_6 \end{array}$$

letting $x_1 = x_3 = x_5 = x_6 = 0$ yields $x_0 = 2$, $x_2 = 1$, $x_4 = 7 \longrightarrow$ feasible solution of value $w = -2$.

now x_3 enters, ratio test: $x_0 : \frac{2}{5/4} = \frac{8}{5}$, $x_4 : \frac{7}{5/2} = \frac{14}{5} \rightarrow$ thus x_0 leaves, $x_3 = \frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_5 + \frac{3}{5}x_6 - \frac{4}{5}x_0$.

$$\begin{array}{rcl} x_2 &=& \frac{11}{5} + \frac{3}{5}x_1 + \frac{2}{5}x_5 + \frac{1}{5}x_6 - \frac{3}{5}x_0 \\ x_3 &=& \frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_5 + \frac{3}{5}x_6 - \frac{4}{5}x_0 \\ x_4 &=& 3 - x_1 - x_6 + 2x_0 \\ \hline w &=& - x_0 \end{array}$$

Feasible solution $x_0 = x_1 = x_5 = x_6 = 0$, $x_2 = \frac{11}{5}$, $x_3 = \frac{8}{5}$, $x_4 = 3$ of value $w = 0$

Since $x_0 = 0$, the solution is also feasible in the original problem.

Now drop the variable x_0 and remove the auxiliary objective w

$$\begin{aligned} x_2 &= \frac{11}{5} + \frac{3}{5}x_1 + \frac{2}{5}x_5 + \frac{1}{5}x_6 \\ x_3 &= \frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_5 + \frac{3}{5}x_6 \\ x_4 &= 3 - x_1 - x_6 \end{aligned}$$

Finally, introduce the original objective $z = x_1 - x_2 + x_3$

Note that x_2 and x_3 appear in z but are not non-basic variables of the dictionary

$\rightarrow$ we must substitute them using the dictionary

$$z = x_1 - x_2 + x_3 = x_1 - (\overbrace{\tfrac{11}{5} + \tfrac{3}{5}x_1 + \tfrac{2}{5}x_5 + \tfrac{1}{5}x_6}^{x_2}) + (\overbrace{\tfrac{8}{5} - \tfrac{1}{5}x_1 + \tfrac{1}{5}x_5 + \tfrac{3}{5}x_6}^{x_3}) = -\tfrac{3}{5} + \tfrac{1}{5}x_1 - \tfrac{1}{5}x_5 + \tfrac{2}{5}x_6$$

Thus the resulting starting feasible dictionary for the original problem is as follows:

$$\begin{aligned} x_2 &= \frac{11}{5} + \frac{3}{5}x_1 + \frac{2}{5}x_5 + \frac{1}{5}x_6 \\ x_3 &= \frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_5 + \frac{3}{5}x_6 \\ x_4 &= 3 - x_1 - x_6 \\ \hline z &= -\frac{3}{5} + \frac{1}{5}x_1 - \frac{1}{5}x_5 + \frac{2}{5}x_6 \end{aligned}$$

5 Linear Algebra Review

scalar = a number , could be **real** like $\pi = 3.14...$, **rational** (fraction) like $\frac{3}{4}$, **integer** (whole) like 5 or -6

(scalars "change scale" in proportion to their value)

vector = a sequence of numbers, for example $\mathbf{x} = (3, 1, 0, 2)$

sometimes we write $\mathbf{x} = \begin{bmatrix} 3 & 1 & 0 & 2 \end{bmatrix}$ and say it is a **row vector**,

or we write $\mathbf{x} = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 2 \end{bmatrix}$ and say it is a **column vector**

multiplying a vector by a scalar ("scaling")

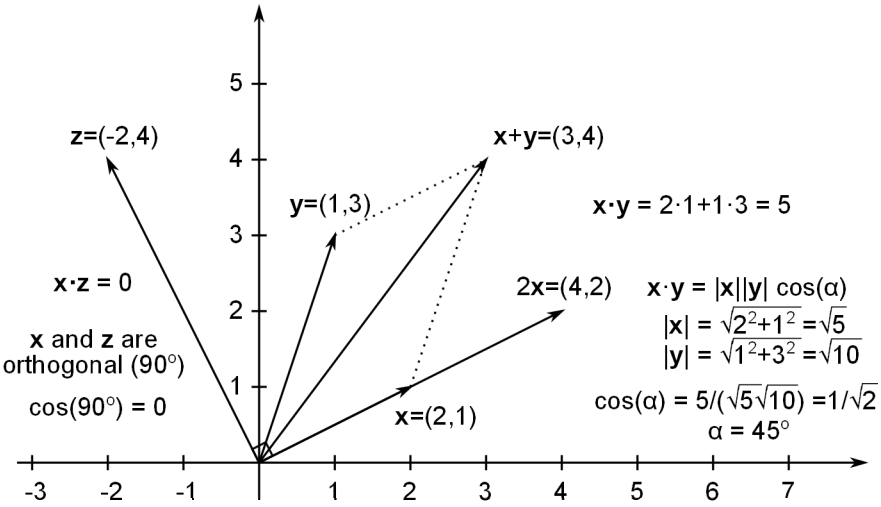
$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \qquad a \text{ is a scalar} \qquad a \cdot \mathbf{x} = (ax_1, ax_2, \ldots, ax_n)$$

For example if $\mathbf{x} = (3, 1, 0, 2)$, then $5 \cdot \mathbf{x} = (5 \cdot 3, 5 \cdot 1, 5 \cdot 0, 5 \cdot 2) = (15, 5, 0, 10)$

adding vectors

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \quad \mathbf{y} = (y_1, y_2, \ldots, y_n) \qquad \mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$$

For example if $\mathbf{x} = (3, 1, 0, 2)$ and $\mathbf{y} = (0, -1, 2, 1)$, then $\mathbf{x} + \mathbf{y} = (3 + 0, 1 + (-1), 0 + 2, 2 + 1) = (3, 0, 2, 3)$



multiplying vectors = **scalar product** ("dot" product) of vectors

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \quad \mathbf{y} = (y_1, y_2, \ldots, y_n) \qquad \mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

For example if $\mathbf{x} = (3, 1, 0, 2)$ and $\mathbf{y} = (0, -1, 2, 1)$, then $\mathbf{x} \cdot \mathbf{y} = 3 \cdot 0 + 1 \cdot (-1) + 0 \cdot 2 + 2 \cdot 1 = 1$

scalar product = corresponds to the **angle** between the vectors

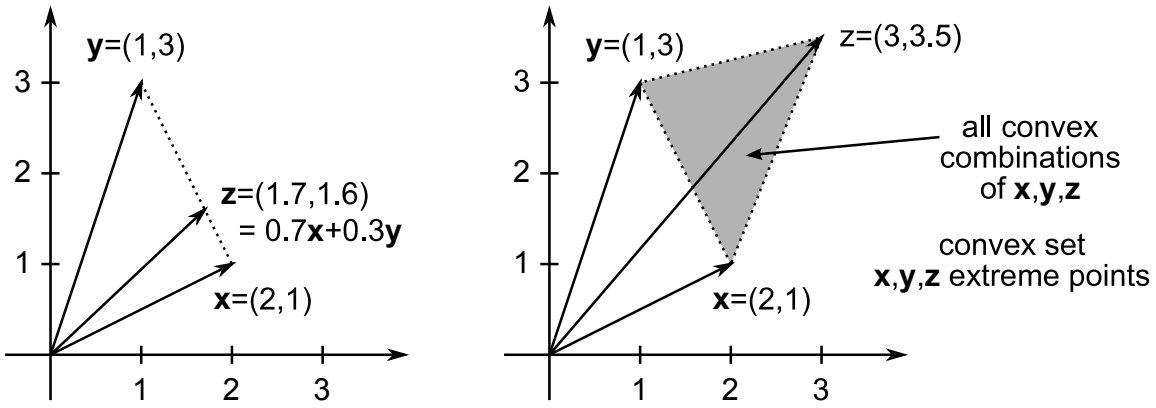
(more precisely, scalar product is **proportional** to the **cosine** of the angle — for instance, the scalar product equals **zero** if and only if the two vector are **orthogonal** (perpenticular) to each other — the angle between them is $90°$)

linear combination of vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m$ is

$$a_1 \mathbf{x}^1 + a_2 \mathbf{x}^2 + \ldots + a_m \mathbf{x}^m$$

where $a_1, a_2, \ldots, a_m$ are numbers (scalars). If each a_i is between 0 and 1, and if $a_1 + a_2 + \cdots + a_m = 1$, then this is called a **convex combination**.

For example, if $\mathbf{x} = (3, 1, 0, 2)$ and $\mathbf{y} = (0, -1, 2, 1)$, then $0.2\mathbf{x} + 0.8\mathbf{y} = (0.6, 0.2, 0, 0.4) + (0, -0.8, 1.6, 0.8) = (0.6, -0.6, 1.6, 1.2)$ is a linear combination of $\mathbf{x}$ and $\mathbf{y}$. Moreover, it is a convex combination, since the coefficients are 0.2 and 0.8 (both between 0 and 1) and $0.2 + 0.8 = 1$.



convex set = is a set of vectors such that whenever we take a convex combination of vectors from this set, then this convex combination also belongs to the set

extreme point of a convex set = cannot be written down as a convex combination of other vectors

(every convex set is the set of all convex combininations (**convex hull**) of its extreme points)

matrix = 2-dimensional array of numbers, for example $\mathbf{A} = \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \\ 0 & 4 & 1 & 2 \end{bmatrix}$

$m \times n$ matrix has m rows and n columns, entries of matrix $\mathbf{A}$ are a_{ij} where i is row and j is column

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \qquad \underbrace{\begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{bmatrix}}_{i\text{-th row of } \mathbf{A}} \qquad \left.\begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}\right\} j\text{-th column of } \mathbf{A}$$

multiplying by a scalar

matrix $\mathbf{A}$ with entries a_{ij}, then $B = k \cdot \mathbf{A}$ is a matrix B with entries $b_{ij} = k \cdot a_{ij}$

$$2 \cdot \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \\ 0 & 4 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 6 & 2 \\ 6 & 4 & 8 & 0 \\ 4 & 6 & 0 & 2 \\ 0 & 8 & 2 & 4 \end{bmatrix}$$

adding matrices of the same size $m \times n$

adding $\mathbf{A}$ with entries a_{ij} to matrix $\mathbf{B}$ with entries b_{ij} is a matrix $\mathbf{C}$ with entries $c_{ij} = a_{ij} + b_{ij}$

$$\begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \\ 0 & 4 & 1 & 2 \end{bmatrix} + \begin{bmatrix} 3 & 1 & 4 & 3 \\ 2 & 0 & 1 & 3 \\ 0 & 2 & 1 & 4 \\ 3 & 0 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 7 & 4 \\ 5 & 2 & 5 & 3 \\ 2 & 5 & 1 & 5 \\ 3 & 4 & 4 & 6 \end{bmatrix}$$

multiplying matrices: matrix $\mathbf{A}$ of size $m \times n$ multiplied by matrix $\mathbf{B}$ of size $n \times k$ results in a matrix

$\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ of size $m \times k$ with entries c_{ij} where $c_{ij} = b_{j1}a_{1i} + b_{j2}a_{2i} + \ldots + b_{jn}a_{ni}$

c_{ij} is the **scalar product** of i-th row of $\mathbf{A}$ with j-th column of $\mathbf{B}$

$$\begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \\ 0 & 4 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 & 4 & 3 \\ 2 & 0 & 1 & 3 \\ 0 & 2 & 1 & 4 \\ 3 & 0 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 7 & 10 & 19 \\ 13 & 11 & 18 & 31 \\ 15 & 2 & 14 & 19 \\ 14 & 2 & 11 & 24 \end{bmatrix} \qquad (2,3,0,1) \cdot (1,0,2,0) = 2$$

Note that $\mathbf{A} \cdot \mathbf{B}$ is **not** the same as $\mathbf{B} \cdot \mathbf{A}$

$\rightarrow$ except for this, matrix addition and multiplication obey exactly the same laws as numbers

$\rightarrow$ from now on **vector** with m entries is to be treated as a $m \times 1$ matrix (column)

$\rightarrow$ for all practical intents and purposes, we can deal with matrices and vectors just like we deal with numbers

multriplying matrix by a **vector** = just like multiplying two matrices

$$\begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \\ 0 & 4 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 11 \\ 2 \\ 2 \end{bmatrix}$$

transpose of a matrix: rotate an $m \times n$ matrix $\mathbf{A}$ along its main diagonal, the resulting $n \times m$ matrix $\mathbf{A}^{\mathrm{T}}$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \end{bmatrix} \qquad \mathbf{A}^{\mathrm{T}} = \begin{bmatrix} 1 & 3 & 2 \\ 0 & 2 & 3 \\ 3 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 1 \end{bmatrix} \qquad \mathbf{x}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 3 & 1 \end{bmatrix}$$

Note that $(\mathbf{A}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{A}$ and $(\mathbf{A} \cdot \mathbf{B})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}} \cdot \mathbf{A}^{\mathrm{T}}$

5.1 Systems of linear equations

A system of linear equations has the following form

$$\begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 \\ \vdots & & & & \ddots & & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & b_m \end{array}$$

Using the matrix notation we can simply write it as $\mathbf{Ax} = \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Basis of solutions to the system $\mathbf{Ax} = \mathbf{b}$

$$\begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}$$

Let us multiply (from the left) both sides of the equation by this matrix: $\begin{bmatrix} -4 & 3 & -2 \\ 8/3 & -2 & 5/3 \\ 5/3 & -1 & 2/3 \end{bmatrix}$

$$\underbrace{\begin{bmatrix} -4 & 3 & -2 \\ 8/3 & -2 & 5/3 \\ 5/3 & -1 & 2/3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \end{bmatrix}} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \underbrace{\begin{bmatrix} -4 & 3 & -2 \\ 8/3 & -2 & 5/3 \\ 5/3 & -1 & 2/3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}}$$

$$\begin{bmatrix} 1 & 0 & 0 & -6 \\ 0 & 1 & 0 & 13/3 \\ 0 & 0 & 1 & 7/3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5 \\ -10/3 \\ -4/3 \end{bmatrix}$$

This operation does not change the solutions to this system (if we multiply with a **non-singular** matrix)

We can expand it back to the system of linear equations

$$\begin{aligned} x_1 \quad - \quad 6x_4 &= 5 \\ x_2 \quad + \quad \tfrac{13}{3}x_4 &= -\tfrac{10}{3} \\ x_3 \; + \; \tfrac{7}{3}x_4 &= -\tfrac{4}{3} \end{aligned} \qquad \rightarrow\rightarrow \qquad \begin{aligned} x_1 &= 5 + 6x_4 \\ x_2 &= -\tfrac{10}{3} - \tfrac{13}{3}x_4 \\ x_3 &= -\tfrac{4}{3} - \tfrac{7}{3}x_4 \end{aligned}$$

The system on the right is in a **dictionary** form.

We can read-off a solution by setting x_4 to some number and calculating the values of x_1, x_2, x_3 from the equations. In particular, we can set $x_4 = 0$ in which case $x_1 = 5$, $x_2 = -\frac{10}{3}$, and $x_3 = -\frac{4}{3}$.

How did we choose the matrix to multiply? We chose the **inverse** matrix of the first three columns.

This is so that the first three columns will be turned to the **identity** matrix $\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Let us write $\mathbf{B}$ for the matrix of first three columns: $\mathbf{B} = \begin{bmatrix} 1 & 0 & 3 \\ 3 & 2 & 4 \\ 2 & 3 & 0 \end{bmatrix}$

inverse matrix of a matrix $\mathbf{B}$ is a matrix $\mathbf{B}^{-1}$ such that $\mathbf{B}^{-1} \cdot \mathbf{B} = \mathbf{I}$

(the inverse matrix may not always exist – we shall ignore this issue here)

How do we obtain an inverse matrix? we perform elementary row operations on the following matrix

$$\underbrace{\left[\begin{array}{ccc} 1 & 0 & 3 \\ 3 & 2 & 4 \\ 2 & 3 & 0 \end{array}\right.}_{\mathbf{B}} \underbrace{\left.\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]}_{\mathbf{I}}$$

elementary row operations

- multiply a row by a non-zero number
- add a row to another row
- exchange two rows

We call $\mathbf{B}$ the **basis matrix**.

Let us try a different basis. Choose the 2nd, 3rd, and 4th columns of $\mathbf{A}$ corresponding to variables x_2, x_3, x_4. (With a slight abuse of notation, we say that $\{x_2, x_3, x_4\}$ is our **basis**, and x_2, x_3, x_4 are **basic variables**.)

$$\mathbf{B} = \begin{bmatrix} 0 & 3 & 1 \\ 2 & 4 & 0 \\ 3 & 0 & 1 \end{bmatrix} \qquad \text{attach the identity matrix:} \qquad \left[\begin{array}{ccc|ccc} 0 & 3 & 1 & 1 & 0 & 0 \\ 2 & 4 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

elementary row operations: multiply the 3rd row by $-2/3$ then add to the 2nd row

$$\left[\begin{array}{ccc|ccc} 0 & 3 & 1 & 1 & 0 & 0 \\ 2 & 4 & 0 & 0 & 1 & 0 \\ -2 & 0 & -\frac{2}{3} & 0 & 0 & -\frac{2}{3} \end{array}\right] \qquad \left[\begin{array}{ccc|ccc} 0 & 3 & 1 & 1 & 0 & 0 \\ 0 & 4 & -\frac{2}{3} & 0 & 1 & -\frac{2}{3} \\ -2 & 0 & -\frac{2}{3} & 0 & 0 & -\frac{2}{3} \end{array}\right]$$

multiply the 1st row by $-4/3$ and then add to the 2nd row

$$\left[\begin{array}{ccc|ccc} 0 & -4 & -\frac{4}{3} & -\frac{4}{3} & 0 & 0 \\ 0 & 4 & -\frac{2}{3} & 0 & 1 & -\frac{2}{3} \\ -2 & 0 & -\frac{2}{3} & 0 & 0 & -\frac{2}{3} \end{array}\right] \qquad \left[\begin{array}{ccc|ccc} 0 & -4 & -\frac{4}{3} & -\frac{4}{3} & 0 & 0 \\ 0 & 0 & -2 & -\frac{4}{3} & 1 & -\frac{2}{3} \\ -2 & 0 & -\frac{2}{3} & 0 & 0 & -\frac{2}{3} \end{array}\right]$$

multiply the 1st, 2nd, 3rd row by $-1/4, 1/6, -1/2$ respectively; then add 2nd row to 1st and 3rd row

$$\left[\begin{array}{ccc|ccc} 0 & 1 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & -\frac{1}{3} & -\frac{2}{9} & \frac{1}{6} & -\frac{1}{9} \\ 1 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} \end{array}\right] \qquad \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & \frac{1}{9} & \frac{1}{6} & -\frac{1}{9} \\ 0 & 0 & -\frac{1}{3} & -\frac{2}{9} & \frac{1}{6} & -\frac{1}{9} \\ 1 & 0 & 0 & -\frac{2}{9} & \frac{1}{6} & \frac{2}{9} \end{array}\right]$$

multiply the 2nd row by $-1/2$ and swap the order of rows to get the identity matrix on the left

$$\left[\begin{array}{ccc|ccc} 0 & 1 & 0 & \frac{1}{9} & \frac{1}{6} & -\frac{1}{9} \\ 0 & 0 & 1 & \frac{2}{3} & -\frac{1}{2} & \frac{1}{3} \\ 1 & 0 & 0 & -\frac{2}{9} & \frac{1}{6} & \frac{2}{9} \end{array}\right] \qquad \underbrace{\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -\frac{2}{9} & \frac{1}{6} & \frac{2}{9} \\ 0 & 1 & 0 & \frac{1}{9} & \frac{1}{6} & -\frac{1}{9} \\ 0 & 0 & 1 & \frac{2}{3} & -\frac{1}{2} & \frac{1}{3} \end{array}\right]}_{\mathbf{I}\qquad\qquad\qquad\qquad\qquad\qquad\mathbf{B}^{-1}}$$

Going back to our original system $\mathbf{Ax} = \mathbf{b}$, we multiply by $\mathbf{B}^{-1}$ from the left

$$\mathbf{B}^{-1}\mathbf{Ax} = \mathbf{B}^{-1}\mathbf{b}$$

$$\underbrace{\begin{bmatrix} -\frac{2}{9} & \frac{1}{6} & \frac{2}{9} \\ \frac{1}{9} & \frac{1}{6} & -\frac{1}{9} \\ \frac{2}{3} & -\frac{1}{2} & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3 & 2 & 4 & 0 \\ 2 & 3 & 0 & 1 \end{bmatrix}}_{\begin{bmatrix} \frac{13}{18} & 1 & 0 & 0 \\ \frac{7}{18} & 0 & 1 & 0 \\ -\frac{1}{6} & 0 & 0 & 1 \end{bmatrix}} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{2}{9} & \frac{1}{6} & \frac{2}{9} \\ \frac{1}{9} & \frac{1}{6} & -\frac{1}{9} \\ \frac{2}{3} & -\frac{1}{2} & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}}_{\begin{bmatrix} \frac{5}{18} \\ \frac{11}{18} \\ -\frac{5}{6} \end{bmatrix}}$$

$$\begin{bmatrix} \frac{13}{18} & 1 & 0 & 0 \\ \frac{7}{18} & 0 & 1 & 0 \\ -\frac{1}{6} & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{5}{18} \\ \frac{11}{18} \\ -\frac{5}{6} \end{bmatrix}$$

$$\begin{array}{rcl} \frac{13}{18}x_1 + x_2 = \frac{5}{18} & & x_2 = \frac{5}{18} - \frac{13}{18}x_1 \\ \frac{7}{18}x_1 + x_3 = \frac{11}{18} & \rightarrow\rightarrow & x_3 = \frac{11}{18} - \frac{7}{18}x_1 \\ -\frac{1}{6}x_1 + x_4 = -\frac{5}{6} & & x_4 = -\frac{5}{6} + \frac{1}{6}x_1 \end{array}$$

5.2 Summary

Consider the maximization problem

$$\begin{array}{c} \max \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{array} \qquad \begin{array}{l} \max \quad \mathbf{c}_B^{\mathrm{T}}\mathbf{x}_B + \mathbf{c}_N^{\mathrm{T}}\mathbf{x}_N \\ \text{subject to } \left[\begin{array}{c|c} \mathbf{B} & \mathbf{N} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{array}\right] = \left[\begin{array}{c} \mathbf{b} \end{array}\right] \end{array}$$

1. choose basic variables, let $\mathbf{x}_B$ denote the **vector** of basic variables
2. let $\mathbf{B}$ denote the **basis** matrix formed by taking the columns of $\mathbf{A}$ corresponding to the basic variables
3. let $\mathbf{c}_B$ denote the vector of the coefficients of $\mathbf{c}$ of the basic variables
4. let $\mathbf{x}_N$ denote the vector of the remaining (non-basic) variables, and let $\mathbf{c}_N$ denote the vector of the corresponding coefficients of $\mathbf{c}$
5. let $\mathbf{N}$ denote the columns of $\mathbf{A}$ corresponding to the non-basic variables in $\mathbf{x}_N$

Then (assuming $\mathbf{B}^{-1}$ exists) we can rewrite

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N &= \mathbf{b} \\ \mathbf{B}^{-1}(\mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N) &= \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{B}^{-1}\mathbf{B}\mathbf{x}_B + \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N &= \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{x}_B + \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N &= \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{x}_B &= \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \end{aligned}$$

Now we can substitute $\mathbf{x}_B$ to the objective function:

$$z = \mathbf{c}^{\mathrm{T}}\mathbf{x} = \mathbf{c}_B^{\mathrm{T}}\mathbf{x}_B + \mathbf{c}_N^{\mathrm{T}}\mathbf{x}_N = \mathbf{c}_B^{\mathrm{T}}\Big(\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N\Big) + \mathbf{c}_N^{\mathrm{T}}\mathbf{x}_N = \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{b} + \Big(\mathbf{c}_N^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{N}\Big)\mathbf{x}_N$$

We put it together to obtain the corresponding dictionary:

$$\begin{array}{rcl} \mathbf{x}_B &=& \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \\ \hline z &=& \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{b} + \Big(\mathbf{c}_N^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{N}\Big)\mathbf{x}_N \end{array}$$

From this we immediately see that the corresponding **basic solution** when $\mathbf{x}_N = 0$ is given as

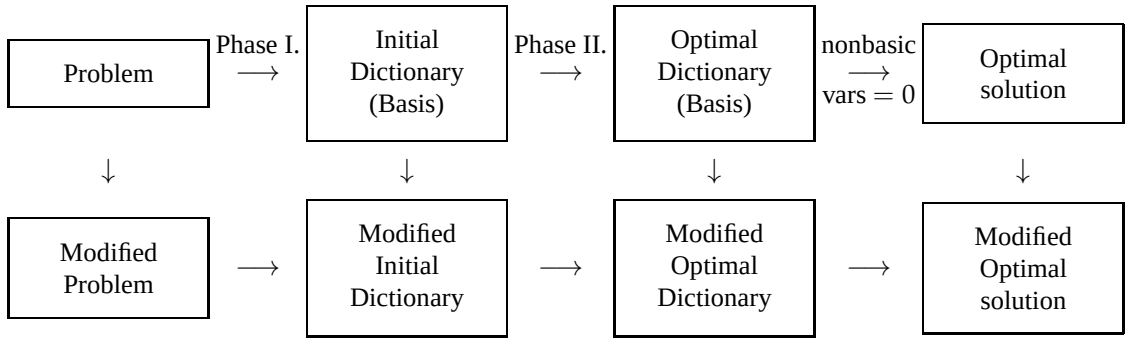
$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \qquad \text{with the value of the objective function} \qquad z = \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{b}$$

The non-basic variables in the objective function are $\Big(\mathbf{c}_N^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{N}\Big)\mathbf{x}_N$. Their coefficients tell us whether or not this solution is optimal. In other words, the solution is optimal(maximal) if $\Big(\mathbf{c}_N^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{N}\Big) \leq 0$.

6 Sensitivity Analysis

Also called *post-optimality analysis*: determining what effect (if any) do changes to the input problem have on the optimality (feasibility) of a solution to the problem.

Motivation: often coefficients/values in a mathematical formulation are only best estimates; we need to know how much room for an error do we have, how sensitive the solution is to the quality of these estimates.



Modified problem is obtained by:

- changing the objective function
- changing the right-hand side (rhs) of a constraint
- adding a variable/activity
- adding a constraint

For what changes is the original optimal solution also optimal in the modified problem?

For what changes is it feasible?

For what changes is the optimal basis also optimal in the modified problem?

How do we recompute modified optimal solution from optimal basis?

How do we recompute modified optimal dictionary?

Key concepts:

- *shadow prices* = a mechanism to assign prices to *items* (rows)
- *reduced costs* = costs of *activities* (columns) in terms of shadow prices

(We will come back to these later; for now, let us look at examples.)

Matrix notation

$$\begin{array}{lrcrcrcl} \max & c_1x_1 & + & c_2x_2 & + & \ldots & + & c_nx_n \\ \text{subject to} & a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 \\ & a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 \\ & \vdots & & \vdots & & & & \vdots & & \vdots \\ & a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & b_m \\ & & & & & & & x_1, x_2, \ldots, x_n & \geq & 0 \end{array}$$

In matrix notation, we express it as follows:

$$\begin{array}{l} \max \ \mathbf{cx} \\ \quad \mathbf{Ax} = \mathbf{b} \\ \quad x \geq 0 \end{array} \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \ \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \ \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \qquad \mathbf{c} = (c_1, c_2, \ldots, c_n)$$

When we pick a basis, we split variables $\mathbf{x}$ into *basic* $\mathbf{x}_B$ variables and *non-basic* $\mathbf{x}_N$. Same way we split $\mathbf{c}$ into $\mathbf{c}_B$ and $\mathbf{c}_N$, and the matrix $\mathbf{A}$ splits into $\mathbf{B}$ and $\mathbf{N}$, where $\mathbf{B}$ are the columns corresponding to basic variables, and $\mathbf{N}$ the columns of non-basic variables. Then we can rewrite the LP as follows:

$$\begin{array}{l} \max \ \mathbf{c}_B\mathbf{x}_B + \mathbf{c}_N\mathbf{x}_N \\ \quad \mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{b} \\ \quad \mathbf{x} \geq 0 \end{array}$$

Assuming that $\mathbf{B}$ is non-singular, we have the inverse $\mathbf{B}^{-1}$ of $\mathbf{B}$ and so we can express $\mathbf{x}_B$ by multiplying the equation by $\mathbf{B}^{-1}$ from the left.

$$\begin{array}{l} \max \ \mathbf{c}_B\mathbf{x}_B + \mathbf{c}_N\mathbf{x}_N \\ \quad \mathbf{x}_B + \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N = \mathbf{B}^{-1}\mathbf{b} \\ \quad \mathbf{x} \geq 0 \end{array}$$

Now we substitute $\mathbf{x}_B$ into the objective function.
(We can then express the corresponding dictionary.)

$$\begin{array}{l} \max \ \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{N})\mathbf{x}_N \\ \quad \mathbf{x}_B + \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N = \mathbf{B}^{-1}\mathbf{b} \\ \quad \mathbf{x} \geq 0 \end{array} \qquad \begin{array}{r} \mathbf{x}_B = \quad \mathbf{B}^{-1}\mathbf{b} - \quad \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \\ \hline z = \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{N})\mathbf{x}_N \end{array}$$

From this we can derive all important quantities:

value of the *objective function*:	$\mathbf{c}_B\mathbf{B}^{-1}\mathbf{b}$
values of the *basic variables*:	$\mathbf{B}^{-1}\mathbf{b}$
shadow prices:	$\mathbf{c}_B\mathbf{B}^{-1}$
reduced costs:	$\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{N}$

Example

Let us illustrate this on our favourite example. We use basis $\{x_1, x_2, x_5\}$.

$$\begin{array}{lrcrcrcrcl} \max & 3x_1 & + & 2x_2 \\ & x_1 & + & x_2 & + & x_3 & & & = & 80 \\ & 2x_1 & + & x_2 & & & + & x_4 & = & 100 \\ & x_1 & & & & & & + \ x_5 & = & 40 \\ & & & & & x_1, x_2, x_3, x_4, x_5 & & & \geq & 0 \end{array}$$

$$\mathbf{c}_B = (3,2,0) \quad \mathbf{c}_N = (0,0)$$

$$\mathbf{x}_B = \begin{pmatrix} x_1 \\ x_2 \\ x_5 \end{pmatrix} \quad \mathbf{x}_N = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 80 \\ 100 \\ 40 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad \mathbf{N} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

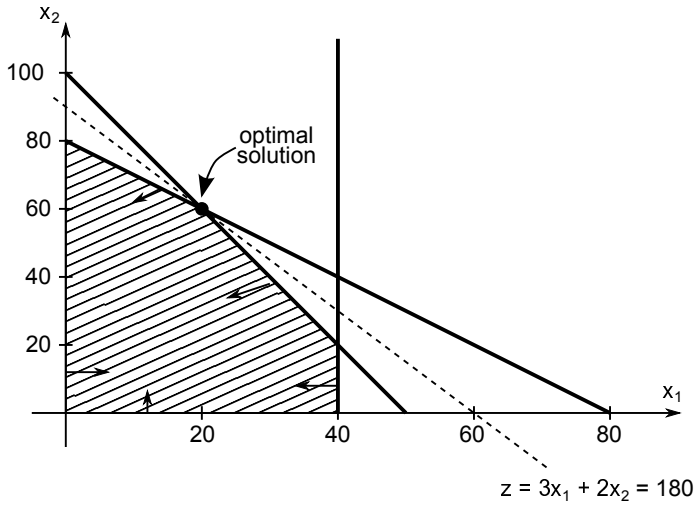
the inverse of B is as follows: $\mathbf{B}^{-1} = \begin{pmatrix} -1 & 1 & 0 \\ 2 & -1 & 0 \\ 1 & -1 & 1 \end{pmatrix}$

basic variables: $\mathbf{B}^{-1}\mathbf{b} = \begin{pmatrix} -1 & 1 & 0 \\ 2 & -1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 80 \\ 100 \\ 40 \end{pmatrix} = (-80+100, 160-100, 80-100+40) = (20, 60, 20)$

shadow prices: $\boldsymbol{\pi} = \mathbf{c}_B\mathbf{B}^{-1} = (3,2,0) \begin{pmatrix} -1 & 1 & 0 \\ 2 & -1 & 0 \\ 1 & -1 & 1 \end{pmatrix} = (-3+4+0, 3-2+0, 0+0+0) = (1,1,0)$

reduced costs: $\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{N} = \mathbf{c}_n - \boldsymbol{\pi}\mathbf{N} = (0,0) - (1,1,0) \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} = (0,0) - (1,1) = (-1,-1)$

$$\begin{array}{rrcl} \text{Max} & 3x_1 + 2x_2 & & \\ & x_1 + x_2 & \leq & 80 \\ & 2x_1 + x_2 & \leq & 100 \\ & x_1 & \leq & 40 \\ & x_1, x_2 & \geq & 0 \end{array}$$

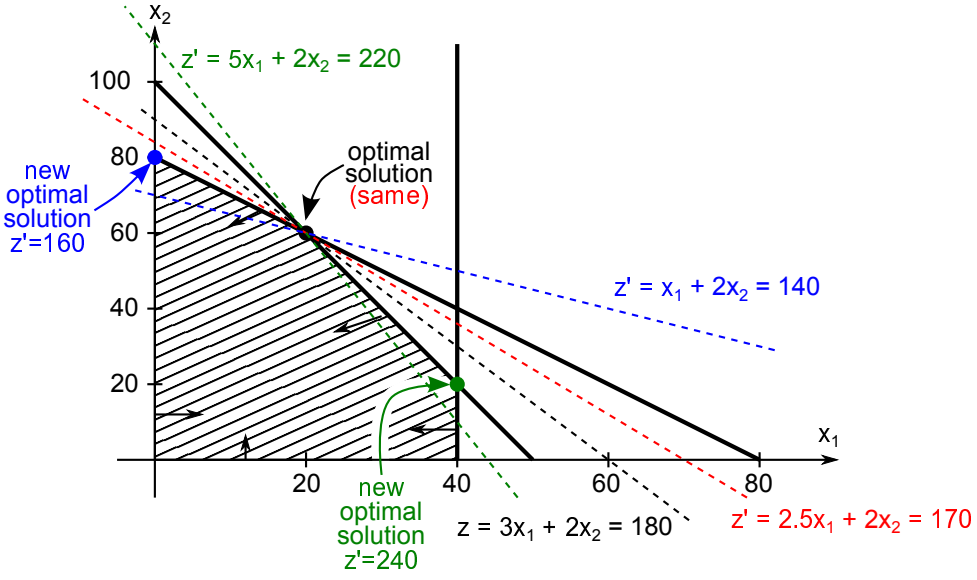


6.1 Changing the objective function

optimal solution $x_1 = 20$, $x_2 = 60$

change the coefficient $c_1 = 3$ in z to 2.5 $\longrightarrow z' = 2.5x_1 + 2x_2$

$\longrightarrow$ the solution $x_1 = 20$, $x_2 = 60$ still optimal, value $z' = 170$



change $c_1 = 3$ to 1 $\longrightarrow z' = x_1 + 2x_2$

$\longrightarrow$ the solution $x_1 = 20$, $x_2 = 60$ **not optimal**, value $z' = 140$

$\longrightarrow$ better solution $x_1 = 0$, $x_2 = 80$, value $z' = 160 > 140$

change $c_1 = 3$ to $5 \longrightarrow z' = 5x_1 + 2x_2$
$\longrightarrow$ the solution $x_1 = 20$, $x_2 = 60$ **not optimal**, value $z' = 220$
$\longrightarrow$ better solution $x_1 = 40$, $x_2 = 20$, value $z' = 240 > 220$

problem formulation

$$\begin{aligned} \text{Max } & 3x_1 + 2x_2 \\ & x_1 + x_2 \le 80 \\ & 2x_1 + x_2 \le 100 \\ & x_1 \le 40 \\ & x_1, x_2 \ge 0 \end{aligned}$$

initial (feasible) dictionary

$$\begin{aligned} x_3 &= 80 - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned}$$

optimal dictionary

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned}$$

From any dictionary (basic vars above the line) we can always express the objective function below the line

$$z = 3x_1 + 2x_2 = 3(\overbrace{20 + x_3 - x_4}^{x_1}) + 2(\overbrace{60 - 2x_3 + x_4}^{x_2}) = 60 + 3x_3 - 3x_4 + 120 - 4x_3 + 2x_4 = 180 - x_3 - x_4$$

This allows us to easily obtain a dictionary for the modified problem corresponding to $x_1 = 20$, $x_2 = 60$

$c_1 = 2.5$

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ \hline z' &= 2.5x_1 + 2x_2 \\ &= 50 + 2.5x_3 - 2.5x_4 \\ &+ 120 - 4x_3 + 2x_4 \\ z' &= 170 - 1.5x_3 - 0.5x_4 \end{aligned}$$

optimal

$c_1 = 1$

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ \hline z' &= x_1 + 2x_2 \\ &= 20 + x_3 - x_4 \\ &+ 120 - 4x_3 + 2x_4 \\ z' &= 140 - 3x_3 + x_4 \end{aligned}$$

not optimal

$c_1 = 5$

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ \hline z' &= 5x_1 + 2x_2 \\ &= 100 + 5x_3 - 5x_4 \\ &+ 120 - 4x_3 + 2x_4 \\ z' &= 220 + x_3 - 3x_4 \end{aligned}$$

not optimal

Coefficient ranging

we want to find the range for the coefficient c_1 for which the optimal solution remains optimal

$$\begin{aligned} z' = c_1x_1 + 2x_2 &= c_1(\overbrace{20 + x_3 - x_4}^{x_1}) + 2(\overbrace{60 - 2x_3 + x_4}^{x_2}) = \\ &= 20c_1 + c_1x_3 - c_1x_4 + 120 - 4x_3 + 2x_4 \\ &= (20c_1 + 120) + (\underbrace{c_1 - 4}_{\le 0})x_3 + (\underbrace{2 - c_1}_{\le 0})x_4 \end{aligned}$$

It is **optimal** if all coefficients are non-positive $\longrightarrow$ $c_1 - 4 \le 0$ and $2 - c_1 \le 0 \implies$ $\boxed{2 \le c_1 \le 4}$

Same for 2nd coefficient c_2:

$$\begin{aligned} z' = 3x_1 + c_2x_2 &= 3(\overbrace{20 + x_3 - x_4}^{x_1}) + c_2(\overbrace{60 - 2x_3 + x_4}^{x_2}) = \\ &= 60 + 3x_3 - 3x_4 + 60c_2 - 2c_2x_3 + c_2x_4 \\ &= (60c_2 + 60) + (\underbrace{3 - 2c_2}_{\le 0})x_3 + (\underbrace{c_2 - 3}_{\le 0})x_4 \end{aligned}$$

$\longrightarrow$ $3 - 2c_2 \le 0$ and $c_2 - 3 \le 0 \implies$ $\boxed{1.5 \le c_2 \le 3}$

General formula

Changing the coefficient c_i of x_i in z to $(c_i + \Delta) \rightarrow$ the value of z changes by Δx_i

$$z' = \underbrace{\mathbf{c}_B\mathbf{x}_B + \mathbf{c}_N\mathbf{x}_N}_{z} + \Delta x_i = z + \Delta x_i = \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_N - \boldsymbol{\pi}\mathbf{N})\mathbf{x}_N + \Delta x_i$$

Non-basic variable: If x_i is j-th non-basic variable, then only the coefficient of x_i changes in z, it is increased by Δ, we must sure it remains non-positive.

The original coefficient of x_i is the reduced cost $\overline{c_i}$ of x_i, the j-th coefficient in $\mathbf{c}_N - \boldsymbol{\pi}\mathbf{N}$.

$$\overline{c_i} + \Delta \leq 0 \quad \rightarrow \quad \text{solve for } \Delta \quad \rightarrow \quad \boxed{\Delta \leq -\overline{c_i}}$$

Basic variable: If x_i is j-th basic variable, then all coefficients of z change; the coefficients change by $-\Delta\mathbf{a}$ where $\mathbf{a}$ is the j-th row of $\mathbf{B}^{-1}\mathbf{N}$. The resulting coefficients must remain non-positive.

The original coefficients of variables in z are the reduced costs $\mathbf{c}_N - \boldsymbol{\pi}\mathbf{N}$.

$$(\mathbf{c}_N - \boldsymbol{\pi}\mathbf{N}) - \Delta\mathbf{a} \leq 0 \quad \rightarrow \quad \text{solve for } \Delta$$

6.2 Changing the right-hand side value

$$\begin{array}{rcl} \text{Max } 3x_1 + 2x_2 & & \\ x_1 + x_2 & \leq & 80 \\ 2x_1 + x_2 & \leq & 100 \\ x_1 & \leq & 40 \\ x_1, x_2 & \geq & 0 \end{array}$$

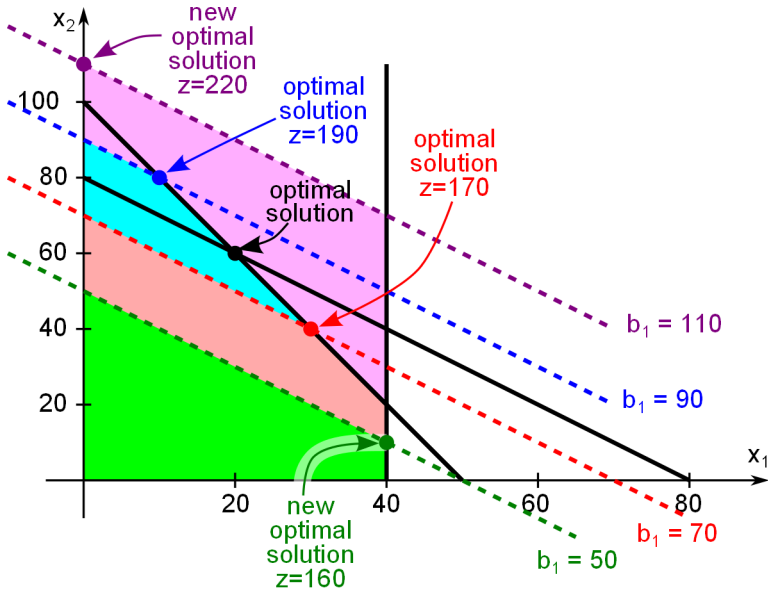
optimal dictionary

$$\begin{array}{rcl} x_1 = 20 + x_3 - x_4 \\ x_2 = 60 - 2x_3 + x_4 \\ x_5 = 20 - x_3 + x_4 \\ \hline z = 180 - x_3 - x_4 \end{array}$$

optimal basis $\{x_1, x_2, x_5\}$

change the coefficient $b_1 = 80$ on the rhs of the 1st constraint to 70 $\longrightarrow x_1 + x_2 \leq 70$
$\longrightarrow$ the point $x_1 = 20$, $x_2 = 60$ **not feasible**
$\longrightarrow$ new optimum $x_1 = 30$, $x_2 = 40$ but **same basis** $\{x_1, x_2, x_5\}$

change b_1 to 50 $\longrightarrow x_1 + x_2 \leq 50$
$\longrightarrow$ the point $x_1 = 20$, $x_2 = 60$ **not feasible**
$\longrightarrow$ new optimum $x_1 = 10$, $x_2 = 40$ **new basis** $\{x_1, x_2, x_4\}$



change b_1 to 90 $\longrightarrow x_1 + x_2 \leq 90$
$\longrightarrow$ the point $x_1 = 20$, $x_2 = 60$ feasible but **not optimal**
$\longrightarrow$ new optimum $x_1 = 10$, $x_2 = 80$ **same basis**

change b_1 to 110 $\longrightarrow x_1 + x_2 \leq 110$
$\longrightarrow$ the point $x_1 = 20$, $x_2 = 60$ feasible but **not optimal**
$\longrightarrow$ new optimum $x_1 = 0$, $x_2 = 110$ **new basis** $\{x_2, x_4, x_5\}$

Coefficient ranging

We want to find the values of b_1 for which the optimal basis remains optimal (optimal solution may not)

initial dictionary

$$\begin{aligned} x_3 &= 80 - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned}$$

final dictionary

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned}$$

Changing $b_1 = 80$ to $80 + \Delta_1$ where Δ_1 can be positive or negative. How does it change the final dictionary?

$$\begin{aligned} x_3 &= (80+\Delta_1) - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned} \qquad \begin{aligned} (x_3-\Delta_1) &= 80 - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned} \qquad \begin{aligned} x_3' &= 80 - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned}$$

where $x_3' = x_3 - \Delta_1$

same dictionary except that x_3 is now $x_3' \longrightarrow$ following the same pivotting steps
we pivot to the basis $\{x_1, x_2, x_5\}$ and must reach the **same** final dictionary (with x_3' in place of x_3)

$$\begin{aligned} x_1 &= 20 + x_3' - x_4 \\ x_2 &= 60 - 2x_3' + x_4 \\ x_5 &= 20 - x_3' + x_4 \\ \hline z &= 180 - x_3' - x_4 \end{aligned} \qquad \begin{aligned} x_1 &= 20 + (x_3-\Delta_1) - x_4 \\ x_2 &= 60 - 2(x_3-\Delta_1) + x_4 \\ x_5 &= 20 - (x_3-\Delta_1) + x_4 \\ \hline z &= 180 - (x_3-\Delta_1) - x_4 \end{aligned} \qquad \begin{aligned} x_1 &= (20-\Delta_1) + x_3 - x_4 \\ x_2 &= (60+2\Delta_1) - 2x_3 + x_4 \\ x_5 &= (20+\Delta_1) - x_3 + x_4 \\ \hline z &= (180+\Delta_1) - x_3 - x_4 \end{aligned}$$

final modified dictionary

When is this dictionary optimal? When it is **feasible**, since all coefficients in z are non-positive

It is feasible, if x_1, x_2, x_5 are non-negative. Setting the non-basic variables $x_3 = x_4 = 0$, we obtain

$$\left.\begin{array}{ll} x_1 = 20 - \Delta_1 \geq 0 & \Delta_1 \leq 20 \\ x_2 = 60 + 2\Delta_1 \geq 0 \quad \longrightarrow & \Delta_1 \geq -30 \\ x_5 = 20 + \Delta_1 \geq 0 & \Delta_1 \geq -20 \end{array}\right\} \quad \boxed{-20 \leq \Delta_1 \leq 20}$$

Try $\Delta_1 = \ \ 10 \rightarrow b_1 = 90$, $x_1 = 10$, $x_2 = 80$, $z = 190 \rightarrow$ **exactly** as we saw before
$\Delta_1 = -10 \rightarrow b_1 = 70$, $x_1 = 30$, $x_2 = 40$, $z = 170$

Similarly for $b_2 = 100$ the coeff of the 2nd constraint $\longrightarrow b_2 = 100 + \Delta_2 \Longrightarrow$ substitute $x_4' = x_4 - \Delta_2$

$$\begin{aligned} x_1 &= 20 + x_3 - x_4' \\ x_2 &= 60 - 2x_3 + x_4' \\ x_5 &= 20 - x_3 + x_4' \\ \hline z &= 180 - x_3 - x_4' \end{aligned} \qquad \begin{aligned} x_1 &= 20 + x_3 - (x_4-\Delta_2) \\ x_2 &= 60 - 2x_3 + (x_4-\Delta_2) \\ x_5 &= 20 - x_3 + (x_4-\Delta_2) \\ \hline z &= 180 - x_3 - (x_4-\Delta_2) \end{aligned} \qquad \begin{aligned} x_1 &= (20+\Delta_2) + x_3 - x_4 \\ x_2 &= (60-\Delta_2) - 2x_3 + x_4 \\ x_5 &= (20-\Delta_2) - x_3 + x_4 \\ \hline z &= (180+\Delta_2) - x_3 - x_4 \end{aligned}$$

optimal if feasible $\longrightarrow$

$$\left.\begin{array}{ll} x_1 = 20 + \Delta_2 \geq 0 & \Delta_2 \geq -20 \\ x_2 = 60 - \Delta_2 \geq 0 \quad \longrightarrow & \Delta_2 \leq 60 \\ x_5 = 20 - \Delta_2 \geq 0 & \Delta_2 \leq 20 \end{array}\right\} \quad \boxed{-20 \leq \Delta_2 \leq 20}$$

Finally, changing $b_3 = 40$, the rhs of the 3rd constraint $\longrightarrow b_3 = 40 + \Delta_3 \Longrightarrow$ substitute $x_5' = x_5 - \Delta_3$

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5' &= 20 - x_3 + x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned} \qquad \begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ (x_5-\Delta_3) &= 20 - x_3 + x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned} \qquad \begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= (20+\Delta_3) - x_3 + x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned}$$

optimal if feasible $\longrightarrow \quad (20+\Delta_3) \geq 0 \quad \longrightarrow \quad \Delta_3 \geq -20 \quad \Longrightarrow \quad \boxed{-20 \leq \Delta_3 \leq \infty}$

General formula

Changing the rhs coefficient b_i to $b_i + \Delta$. Let $\mathbf{d}$ denote the ***i*-the column of** B^{-1}. The dictionary changes:

$$\begin{array}{rrrr} \mathbf{x}_B = & \mathbf{B}^{-1}\mathbf{b} + & \Delta\mathbf{d} - & \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \\ \hline z = \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b} + \mathbf{c}_B\Delta\mathbf{d} + (\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{N})\mathbf{x}_N & & & \end{array}$$

We find for what values of Δ the values of $\mathbf{x}_B$ are non-negative (if the non-basic variables $\mathbf{x}_N$ are set to 0).

$$\mathbf{x}_B = \underbrace{\mathbf{B}^{-1}\mathbf{b}}_{\text{old values}} + \Delta\mathbf{d} \geq 0$$

Shadow prices

Let us sumarize:

changing $b_1 = 80$ to $80 + \Delta_1$	changing $b_2 = 100$ to $100 + \Delta_2$	changing $b_3 = 40$ to $40 + \Delta_3$
$x_1 = 20 - \Delta_1$	$x_1 = 20 + \Delta_2$	$x_1 = 20$
$x_2 = 60 + 2\Delta_1$	$x_2 = 60 - \Delta_2$	$x_2 = 60$
$x_5 = 20 + \Delta_1$	$x_5 = 20 - \Delta_2$	$x_5 = 20 + \Delta_3$
$z = 180 + \Delta_1$	$z = 180 + \Delta_2$	$z = 180$
for $-20 \leq \Delta_1 \leq 20$	for $-20 \leq \Delta_2 \leq 20$	for $-20 \leq \Delta_3 \leq \infty$

if the change is within the above bounds on $\Delta_1, \Delta_2, \Delta_3$, then:

- *increasing* the value of b_1 by **1 unit** *increases* the objective (profit) by \$1 (*decreasing*) (*decreases*)
- increasing/decreasing b_2 by **1 unit** increases/decreases the profit by \$1
- increasing/decreasing b_3 by **1 unit** increases/decreases the profit by \$0

} **shadow prices**

Economic interpretation: if instead of production we sell/rent all capacity (80 hours in the carving shop, 100 hours in the finishing shop, 20 units of remaining demand) at shadow prices (\$1/hour, \$1/hour, \$0/unit), we obtain the same profit ($80 \times \$1 + 100 \times \$1 + 20 \times \$0 = \$180$).

In fact, if we sell at \$1/hour up to 20 hours of capacity in the carving shop (since $\Delta_1 \geq -20$) **or** we sell at \$1 up to 20 hours of capacity in the finishing shop (since $\Delta_2 \geq -20$), and optimally produce toys using the remaining capacity, then our profit does not change (remains \$180). Similarly, if we obtain additional capacity at \$1/hour up to 20 hours in one of the shops (since $\Delta_1 \leq 20$ and $\Delta_2 \leq 20$), then profit remains the same.

Thus if we can obtain additional shop capacity at less than \$1/hour, then it pays to do it (up to 20 hours) and produce more toys using the additional capacity (yields higher profit). On the other hand, if we can sell shop capacity at more than \$1/hour, then it pays to do it instead of using it for production.

6.3 Detailed example

In the following LP, optimal solution is achieved for basis $\{x_2, x_3, x_4\}$

$$\begin{array}{rrrrrrrrrr} \max\ z = & 2x_1 & - & 3x_2 & + & x_3 & & - & x_5 & \\ & x_1 & - & x_2 & + & x_3 & + & x_4 & = & 4 \\ & -2x_1 & + & x_2 & - & x_3 & & + & x_5 & = 1 \\ & & & 2x_2 & + & x_3 & + & x_4 & - & x_5 = 9 \\ & & & & & x_1, x_2, x_3, x_4, x_5 & & & \geq & 0 \end{array}$$

Find ranges for individual coefficients of the objective function and for rhs coefficients.

Using matrices

1. Basis $\{x_2, x_3, x_4\}$ $\quad \rightarrow$ split the coefficients to the basis matrix **B** and non-basic matrix **N**

$\rightarrow$ calculate the inverse of $\mathbf{B}$, denoted $\mathbf{B}^{-1}$ (recall from linear algebra how to do this)

$$\mathbf{B} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 0 \\ 2 & 1 & 1 \end{pmatrix} \qquad \mathbf{B}^{-1} = \begin{pmatrix} -\frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{3} & -1 & \frac{1}{3} \\ 1 & 1 & 0 \end{pmatrix} \qquad \mathbf{N} = \begin{pmatrix} 1 & 0 \\ -2 & -1 \\ 0 & -1 \end{pmatrix} \qquad \begin{array}{l} \mathbf{c}_B = (-3,1,0) \\ \mathbf{c}_N = (2,-1) \end{array}$$

$$\pi = \mathbf{c}_B\mathbf{B}^{-1} = (-3,1,0)\begin{pmatrix} -\frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{3} & -1 & \frac{1}{3} \\ 1 & 1 & 0 \end{pmatrix} = (\tfrac{2}{3}, -1, -\tfrac{2}{3})$$

$$\mathbf{c}_N - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{N} = \mathbf{c}_N - \pi\mathbf{N} = \left[(2,-1) - (\tfrac{2}{3}, -1, -\tfrac{2}{3})\begin{pmatrix} 1 & 0 \\ -2 & 1 \\ 0 & -1 \end{pmatrix}\right] = (-\tfrac{2}{3}, -\tfrac{2}{3})$$

$$\mathbf{B}^{-1}\mathbf{N} = \begin{pmatrix} -\frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{3} & -1 & \frac{1}{3} \\ 1 & 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ -2 & 1 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{3} & -\frac{1}{3} \\ \frac{5}{3} & -\frac{4}{3} \\ -1 & 1 \end{pmatrix}$$

$$\mathbf{B}^{-1}\mathbf{b} = \begin{pmatrix} -\frac{1}{3} & 0 & \frac{1}{3} \\ -\frac{1}{3} & -1 & \frac{1}{3} \\ 1 & 1 & 0 \end{pmatrix}\begin{pmatrix} 4 \\ 1 \\ 9 \end{pmatrix} = \begin{pmatrix} \frac{5}{3} \\ \frac{2}{3} \\ 5 \end{pmatrix}$$

2. Changing the right-hand side

b_1: change from 4 to $4+\Delta$, the 1st column of $\mathbf{B}^{-1}$ is $\begin{pmatrix} -\frac{1}{3} \\ -\frac{1}{3} \\ 1 \end{pmatrix}$

$$\begin{array}{rcl} x_2 = \frac{5}{3} - \frac{1}{3}\Delta \geq 0 & & \Delta \leq 5 \\ x_3 = \frac{2}{3} - \frac{1}{3}\Delta \geq 0 & \rightarrow & \Delta \leq 2 \quad \rightarrow \quad \boxed{-5 \leq \Delta \leq 2} \\ x_4 = 5 + \Delta \geq 0 & & -5 \leq \Delta \end{array}$$

b_2: change from 1 to $1+\Delta$, the 2nd column of $\mathbf{B}^{-1}$ is $\begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$

$$\begin{array}{rcl} x_2 = \frac{5}{3} + 0\Delta \geq 0 & & 0 \leq \frac{5}{3} \\ x_3 = \frac{2}{3} - \Delta \geq 0 & \rightarrow & \Delta \leq \frac{2}{3} \quad \rightarrow \quad \boxed{-5 \leq \Delta \leq \frac{2}{3}} \\ x_4 = 5 + \Delta \geq 0 & & -5 \leq \Delta \end{array}$$

b_3: change from 9 to $9+\Delta$, the 3rd column of $\mathbf{B}^{-1}$ is $\begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \end{pmatrix}$

$$\begin{array}{rcl} x_2 = \frac{5}{3} + \frac{1}{3}\Delta \geq 0 & & -5 \leq \Delta \\ x_3 = \frac{2}{3} + \frac{1}{3}\Delta \geq 0 & \rightarrow & -2 \leq \Delta \quad \rightarrow \quad \boxed{-2 \leq \Delta} \\ x_4 = 5 + 0\Delta \geq 0 & & 0 \leq 5 \end{array}$$

3. Changing the objective function coefficients

x_1 : change from 2 to $2+\Delta \rightarrow z$ changes by Δx_1

x_1 is non-basic with reduced cost $-\frac{2}{3}$ (1st coeff in $\mathbf{c}_N - \pi\mathbf{N}$) $\rightarrow$ coefficient of x_1 in z is $-\frac{2}{3}$

only the coefficient of x_1 changes in $z \rightarrow$ the new coeff is $-\frac{2}{3} + \Delta \rightarrow$ must be non-positive

$$-\tfrac{2}{3} + \Delta \leq 0 \quad \rightarrow \quad \boxed{\Delta \leq \tfrac{2}{3}}$$

x_2 : change from -3 to $-3+\Delta \to z$ changes by Δx_2

x_2 is 1st basic variable, the coeffients of (x_1, x_5) in z change by $-\Delta\mathbf{a}$ where $\mathbf{a}$ is the 1st row of $\mathbf{B}^{-1}\mathbf{N}$
coefficients of (x_1, x_5) in z are the reduced costs $\mathbf{c}_N - \boldsymbol{\pi}\mathbf{N} = (-\frac{2}{3}, -\frac{2}{3})$ and $\mathbf{a} = (-\frac{1}{3}, -\frac{1}{3})$

the resulting coefficients must be non-positive (in order for the basis to remain optimal)

$$\begin{pmatrix} -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix} - \Delta \begin{pmatrix} -\frac{1}{3} \\ -\frac{1}{3} \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \to \quad \begin{matrix} -\frac{2}{3} + \frac{1}{3}\Delta \ \leq \\ 0 \\ -\frac{2}{3} + \frac{1}{3}\Delta \ \leq \\ 0 \end{matrix} \quad \to \quad \boxed{\Delta \leq 2}$$

x_3 : change from 1 to $1+\Delta \to z$ changes by Δx_3, the 2nd basic variable, the 2nd row of $\mathbf{B}^{-1}\mathbf{N} = (\frac{5}{3}, -\frac{4}{3})$

$$\begin{matrix} -\frac{2}{3} - \frac{5}{3}\Delta \ \leq \\ 0 \\ -\frac{2}{3} + \frac{4}{3}\Delta \ \leq \\ 0 \end{matrix} \quad \to \quad \begin{matrix} \Delta \geq -\frac{2}{5} \\ \Delta \leq \frac{1}{2} \end{matrix} \quad \to \quad \boxed{-\tfrac{2}{5} \leq \Delta \leq \tfrac{1}{2}}$$

x_4 : change from 0 to $0+\Delta \to z$ changes by Δx_4, the 3rd basic variable, the 3rd row of $\mathbf{B}^{-1}\mathbf{N} = (-1, 1)$

$$\begin{matrix} -\frac{2}{3} + \Delta \leq 0 \\ -\frac{2}{3} - \Delta \leq 0 \end{matrix} \quad \to \quad \begin{matrix} \Delta \leq \frac{2}{3} \\ \Delta \geq -\frac{2}{3} \end{matrix} \quad \to \quad \boxed{-\tfrac{2}{3} \leq \Delta \leq \tfrac{2}{3}}$$

x_5 : changes from -1 to $-1+\Delta \to z$ changes by Δx_5, non-basic variable, only coeff $-\frac{2}{3}$ of x_5 changes by Δ

$$-\tfrac{2}{3} + \Delta \leq 0 \quad \to \quad \boxed{\Delta \leq \tfrac{2}{3}}$$

Using dictionaries (optional)

1. Construct corresponding dictionary

Change rhs to (artificial) variables b_1, b_2, b_3 and pivot to the basis $\{x_2, x_3, x_4\}$

$$\begin{matrix} x_1 - x_2 + x_3 + x_4 = b_1 \\ -2x_1 + x_2 - x_3 + x_5 = b_2 \\ 2x_2 + x_3 + x_4 - x_5 = b_3 \end{matrix} \qquad \boxed{x_2 = -b_1 + x_1 + x_3 + x_4}$$

$$-2x_1 + (-b_1 + x_1 + x_3 + x_4) - x_3 + x_5 = b_2$$
$$2(-b_1 + x_1 + x_3 + x_4) + x_3 + x_4 - x_5 = b_3$$

$$\Leftarrow \quad \begin{matrix} -x_1 + x_4 + x_5 = b_1 + b_2 \\ 2x_1 + 3x_3 + 3x_4 - x_5 = 2b_1 + b_3 \end{matrix}$$

$\boxed{x_4 = b_1 + b_2 + x_1 - x_5}$

$$2x_1 + 3x_3 + 3(b_1 + b_2 + x_1 - x_5) - x_5 = 2b_1 + b_3$$

$$5x_1 + 3x_3 + - 4x_5 = -b_1 - 3b_2 + b_3 \qquad \boxed{x_3 = -\tfrac{1}{3}b_1 - b_2 + \tfrac{1}{3}b_3 - \tfrac{5}{3}x_1 + \tfrac{4}{3}x_5}$$

substitute back to x_4 and then to x_2:

$x_4 = b_1 + b_2 + x_1 - x_5$ $\qquad$ $x_2 = -b_1 + x_1 + x_3 + x_4 = -b_1 + x_1 + (-\frac{1}{3}b_1 - b_2 + \frac{1}{3}b_3 - \frac{5}{3}x_1 + \frac{4}{3}x_5)$
$+ (b_1 + b_2 + x_1 - x_5) = -\frac{1}{3}b_1 + \frac{1}{3}b_3 + \frac{1}{3}x_1 + \frac{1}{3}x_5$

substitute to $z \to z = 2x_1 - 3x_2 + x_3 - x_5 = 2x_1 - 3(-\frac{1}{3}b_1 + \frac{1}{3}b_3 + \frac{1}{3}x_1 + \frac{1}{3}x_5)$
$+ (-\frac{1}{3}b_1 - b_2 + \frac{1}{3}b_3 - \frac{5}{3}x_1 + \frac{4}{3}x_5) - x_5 = \frac{2}{3}b_1 - b_2 - \frac{2}{3}b_3 - \frac{2}{3}x_1 - \frac{2}{3}x_5$

Resulting dictionary:

$$\begin{matrix} x_2 = -\frac{1}{3}b_1 + \frac{1}{3}b_3 + \frac{1}{3}x_1 + \frac{1}{3}x_5 \\ x_3 = -\frac{1}{3}b_1 - b_2 + \frac{1}{3}b_3 - \frac{5}{3}x_1 + \frac{4}{3}x_5 \\ x_4 = b_1 + b_2 + x_1 - x_5 \\ \hline z = \frac{2}{3}b_1 - b_2 - \frac{2}{3}b_3 - \frac{2}{3}x_1 - \frac{2}{3}x_5 \end{matrix}$$

2. Changing the right-hand side coefficients

b_1 : change $b_1 = 4$ to $4+\Delta$, and let $b_2 = 1$ and $b_3 = 9$ the same. Check if the resulting values of basic variables are non-negative (remember to set the non-basic variables to zero).

$$\begin{array}{lllll} x_2 = -\frac{1}{3}(4+\Delta) + \frac{9}{3} \geq 0 & & \Delta \leq 5 & & \\ x_3 = -\frac{1}{3}(4+\Delta) - 1 + \frac{9}{3} \geq 0 & \rightarrow & \Delta \leq 2 & \rightarrow & \boxed{-5 \leq \Delta \leq 2} \\ x_4 = (4+\Delta) + 1 \geq 0 & & -5 \leq \Delta & & \end{array}$$

b_2 : change $b_2 = 1$ to $1+\Delta$, let $b_1 = 4$ and $b_3 = 9$

$$\begin{array}{lllll} x_2 = -\frac{4}{3} + \frac{9}{3} \geq 0 & & 0 \leq \frac{5}{3} & & \\ x_3 = -\frac{4}{3} - (1+\Delta) + \frac{9}{3} \geq 0 & \rightarrow & \Delta \leq \frac{2}{3} & \rightarrow & \boxed{-5 \leq \Delta \leq \frac{2}{3}} \\ x_4 = 4 + (1+\Delta) \geq 0 & & -5 \leq \Delta & & \end{array}$$

b_3 : change $b_3 = 9$ to $9+\Delta$, let $b_1 = 4$ and $b_2 = 1$

$$\begin{array}{lllll} x_2 = -\frac{4}{3} + \frac{1}{3}(9+\Delta) \geq 0 & & -5 \leq \Delta & & \\ x_3 = -\frac{4}{3} - 1 + \frac{1}{3}(9+\Delta) \geq 0 & \rightarrow & -2 \leq \Delta & \rightarrow & \boxed{-2 \leq \Delta} \\ x_4 = 4 + 1 \geq 0 & & 0 \leq 5 & & \end{array}$$

3. Changing the objective function coefficients

set the rhs back to $b_1 = 4$, $b_2 = 1$, $b_3 = 9$ and express $z = \frac{8}{3} - 1 - \frac{18}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 = -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5$

$\rightarrow$ changing the coefficient of x_i in z by Δ changes z by exactly Δx_i

x_1: change $2x_1$ to $(2+\Delta)x_1 \quad \longrightarrow \quad z' = -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta x_1 = -\frac{13}{3} + \underbrace{(-\frac{2}{3}+\Delta)}_{\leq 0}x_1 - \frac{2}{3}x_5$

Thus $-\frac{2}{3} + \Delta \leq 0 \quad \rightarrow \quad \boxed{\Delta \leq \frac{2}{3}}$

x_2: change $-3x_2$ to $(-3+\Delta)x_2 \quad \longrightarrow \quad z' = -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta x_2$

substitute x_2 from the dictionary

$z' = -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta(\frac{5}{3} + \frac{1}{3}x_1 + \frac{1}{3}x_5) = (-\frac{13}{3} + \Delta\frac{5}{3}) + \underbrace{(-\frac{2}{3}+\frac{1}{3}\Delta)}_{\leq 0}x_1 + \underbrace{(-\frac{2}{3}+\frac{1}{3}\Delta)}_{\leq 0}x_5$

Thus $-\frac{2}{3} + \frac{1}{3}\Delta \leq 0 \quad \rightarrow \quad \boxed{\Delta \leq 2}$

x_3: change $1x_3$ to $(1+\Delta)x_3 \quad \longrightarrow \quad z' = -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta x_3 =$

$= -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta(\frac{2}{3} - \frac{5}{3}x_1 + \frac{4}{3}x_5) = (-\frac{13}{3} + \frac{2}{3}\Delta) + \underbrace{(-\frac{2}{3} - \frac{5}{3}\Delta)}_{\leq 0}x_1 + \underbrace{(-\frac{2}{3} + \frac{4}{3}\Delta)}_{\leq 0}x_5$

$$\begin{array}{lll} -\frac{2}{3} - \frac{5}{3}\Delta \leq 0 & -\frac{2}{5} \leq \Delta & \\ -\frac{2}{3} + \frac{4}{3}\Delta \leq 0 & \Delta \leq \frac{1}{2} & \rightarrow \quad \boxed{-\frac{2}{5} \leq \Delta \leq \frac{1}{2}} \end{array}$$

x_4: change $0x_4$ to $(0+\Delta)x_4 \quad \longrightarrow \quad z' = -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta x_4 =$

$= -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta\,(5 + x_1 - x_5) = (-\frac{13}{3} + 5\Delta) + \underbrace{(-\frac{2}{3} + \Delta)}_{\leq 0}x_1 + \underbrace{(-\frac{2}{3} - \Delta)}_{\leq 0}x_5$

$$\begin{array}{lll} -\frac{2}{3} + \Delta \leq 0 & \Delta \leq \frac{2}{3} & \\ -\frac{2}{3} - \Delta \leq 0 & -\frac{2}{3} \leq \Delta & \rightarrow \quad \boxed{-\frac{2}{3} \leq \Delta \leq \frac{2}{3}} \end{array}$$

x_5: change $-x_5$ to $(-1+\Delta)x_5 \quad \longrightarrow \quad z' = -\frac{13}{3} - \frac{2}{3}x_1 - \frac{2}{3}x_5 + \Delta x_5 = = -\frac{13}{3} - \frac{2}{3}x_1 + \underbrace{(-\frac{2}{3}+\Delta)}_{\leq 0}x_5$

$-\frac{2}{3} + \Delta \leq 0 \quad \rightarrow \quad \boxed{\Delta \leq \frac{2}{3}}$

6.4 Adding a variable/activity

Toy cars: $\frac{1}{2}$ hour carving, 1 hour finishing, sale price \$1

$x_6 =$ the number of cars produced

$$\begin{array}{rrrrrrrrrrrrr}
\text{Max} & 3x_1 & + & 2x_2 & & & & + & x_6 & & \\
 & x_1 & + & x_2 & + & x_3 & & + & \frac{1}{2}x_6 & = & 80 \\
 & 2x_1 & + & x_2 & & + & x_4 & & x_6 & = & 100 \\
 & x_1 & & & & & + & x_5 & & = & 40 \\
 & & & & & & x_1, x_2, x_3, x_4, x_5, x_6 & & & \geq & 0
\end{array}$$

$$\begin{array}{rrrrrrrrr}
x_3 & = & 80 & - & x_1 & - & x_2 & - & \frac{1}{2}x_6 \\
x_4 & = & 100 & - & 2x_1 & - & x_2 & - & x_6 \\
x_5 & = & 40 & - & x_1 & & & & \\
\hline
z & = & 0 & + & 3x_1 & + & 2x_2 & + & x_6
\end{array}
\quad \Longrightarrow \quad
\begin{array}{rrrrrrrrr}
x_1 & = & 20 & + & x_3 & - & x_4 & + & ?x_6 \\
x_2 & = & 60 & - & 2x_3 & + & x_4 & + & ?x_6 \\
x_5 & = & 20 & - & x_3 & + & x_4 & + & ?x_6 \\
\hline
z & = & 180 & - & x_3 & - & x_4 & + & ?x_6
\end{array}$$

We make x_6 non-basic $\rightarrow$ produce no cars $x_6 = 0 \rightarrow$ previously optimal solution remains feasible

Is it also optimal? In other words, does it pay to produce cars? *Price out* the new variable

- **pricing out** an activity/variable/column $\equiv$ evaluating the **cost** of producing one unit of product (activity) x_j in terms of current shadow prices
- **reduced cost** $\equiv$ net contribution of one unit of x_j to the value of the objective function (profit)

$$= \quad \begin{array}{c}\textit{revenue from one unit of } x_j \\ \text{(the coefficient of } x_j \text{ in } z\text{)}\end{array} \quad - \quad \begin{array}{c}\textit{the production cost in shadow prices} \\ \text{(multiply the coeffs in } x_j\text{-th column by the} \\ \text{respective shadow prices and then sum up)}\end{array}$$

  $\rightarrow$ if the reduced cost is positive, then it pays to produce x_j;
  $\rightarrow$ if it is negative, then we are better-off producing other items (which is equivalent to selling all available resources at shadow prices)

Shadow prices: $\pi_1 = \$1$, $\pi_2 = \$1$, $\pi_3 = \$0$
$\rightarrow$ one hour in each shop costs \$1 while each unit of remaining demand (3rd constraint) costs \$0

Price out x_6: $\overbrace{\pi_1 \times \frac{1}{2}}^{\textit{cost of carving}} + \overbrace{\pi_2 \times 1}^{\textit{cost of finishing}} + \pi_3 \times 0 = \$1 \times \frac{1}{2} + \$ \times 1 + \$0 \times 0 = \$1.50$

Reduced cost of x_6: $\overbrace{\$1}^{\textit{sale price}} - \overbrace{\$1.50}^{\textit{production cost}} = -\$0.50$

Answer: it does not pay to produce toy cars, we would be losing \$.50 for each unit (instead of producing other products – shadow prices reflect the value of current production) $\rightarrow$ current solution remains optimal

Note: notice that reduced costs are **precisely** the coefficients of non-basic variables in the last row of dictionary, the z-value. If they are all non-positive, then we know the solution is optimal! Put differently, if reduced costs of all non-basic variables (i.e. the coefficients in z) are not positive, then we do not make more by producing any of those items (making one of them basic and thus non-zero) and so the solution is optimal.

Conclusion from this is that -0.5 is the coefficient of x_6 in the modified final dictionary

$$\begin{array}{rrrrrrrrr}
x_1 & = & 20 & + & x_3 & - & x_4 & + & ?x_6 \\
x_2 & = & 60 & - & 2x_3 & + & x_4 & + & ?x_6 \\
x_5 & = & 20 & - & x_3 & + & x_4 & + & ?x_6 \\
\hline
z & = & 180 & - & x_3 & - & x_4 & - & 0.5\,x_6
\end{array}$$

To determine that the solution is optimal, we **do not** need to know all the other missing coefficients! More on that later...

6.5 Adding a constraint

packaging 150 pieces of cardboard, 1 piece per each toy soldier, 2 pieces per each toy train

new constraint: $x_1 + 2x_2 \leq 150$

introduce a slack variable x_6: $x_6 = 150 - x_1 - 2x_2$

substitute from the final dictionary: $x_6 = 150 - (20 + x_3 - x_4) - 2(60 - 2x_3 + x_4) = 10 + 3x_3 - x_4$

Adding this to the final dictionary yields a feasible dictionary (since the value 10 in the contraint is non-negative) $\rightarrow$ optimal solution remains optimal after adding the constraint

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ x_6 &= 10 + 3x_3 - x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned}$$

What if we only have 130 units of cardboard?

Then the original optimal solution becomes infeasible in the modified problem (the constant term works out to -10 instead of 10) and we need to recalculate (or use the Dual Simplex Method)

Conclusion: so long as we have at least 140 units of cardboard we don't need to change the production plan

6.6 Modifying the left-hand side of a constraint

Equipment shortages cause that toy soldiers require 3 hours in the finishing shop (instead of 2).

Original problem

$$\begin{aligned} \text{Max } & 3x_1 + 2x_2 \\ & x_1 + x_2 \leq 80 \\ & 2x_1 + x_2 \leq 100 \\ & x_1 \leq 40 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Initial dictionary

$$\begin{aligned} x_3 &= 80 - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned}$$

Final dictionary

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 \\ \hline z &= 180 - x_3 - x_4 \end{aligned}$$

Modified problem

$$\begin{aligned} \text{Max } & 3x_1 + 2x_2 \\ & x_1 + x_2 \leq 80 \\ & \mathbf{3}x_1 + x_2 \leq 100 \\ & x_1 \leq 40 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Modified Initial dictionary

$$\begin{aligned} x_3 &= 80 - x_1 - x_2 \\ x_4 &= 100 - \mathbf{3}x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned}$$

$\longrightarrow$?

Rearrange the terms in the dictionary and substitute $x_4' = x_4 + x_1$ in the final dictionary (x_4' in place of x_4)

$$\begin{aligned} \overbrace{x_4' \quad x_3}: \quad & 80 - x_1 - x_2 \\ (x_4 + x_1) &= 100 - 2x_1 - x_2 \\ x_5 &= 40 - x_1 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned} \quad \rightarrow \quad \begin{aligned} x_1 &= 20 + x_3 - x_4' \\ x_2 &= 60 - 2x_3 + x_4' \\ x_5 &= 20 - x_3 + x_4' \\ \hline z &= 180 - x_3 - x_4' \end{aligned} \qquad \begin{aligned} x_1 &= 20 + x_3 - (x_4 + x_1) \\ x_2 &= 60 - 2x_3 + (x_4 + x_1) \\ x_5 &= 20 - x_3 + (x_4 + x_1) \\ \hline z &= 180 - x_3 - (x_4 + x_1) \end{aligned}$$

Eliminate x_1 from the rhs by solving for x_1 from the first equation, and then substituting:

$$\begin{aligned} x_1 &= 20 + x_3 - (x_4 + x_1) \\ 2x_1 &= 20 + x_3 - x_4 \\ x_1 &= 10 + \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4 \end{aligned}$$

$$\begin{aligned} x_2 &= 60 - 2x_3 + (x_4 + x_1) = 60 - 2x_3 + x_4 \\ &\quad + (10 + \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4) = 70 - \tfrac{3}{2}x_3 + \tfrac{1}{2}x_4 \\ x_5 &= 20 - x_3 + (x_4 + x_1) = 20 - x_3 + x_4 \\ &\quad + (10 + \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4) = 30 - \tfrac{1}{2}x_3 + \tfrac{1}{2}x_4 \\ z &= 180 - x_3 - (x_4 + x_1) = 180 - x_3 - x_4 \\ &\quad - (10 + \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4) = 170 - \tfrac{3}{2}x_3 - \tfrac{1}{2}x_4 \end{aligned}$$

$$\begin{aligned} x_1 &= 10 + \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4 \\ x_2 &= 70 - \tfrac{3}{2}x_3 + \tfrac{1}{2}x_4 \\ x_5 &= 30 - \tfrac{1}{2}x_3 + \tfrac{1}{2}x_4 \\ \hline z &= 170 - \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4 \end{aligned}$$

Conclusion: the optimal solution remains optimal.

What if we buy new machine that shortens the finishing process of toy soldiers to 1.5 hour (instead of 2) ? Then current optimal solution will no longer be optimal ($+x_4$ will appear in the z value).

(Curiously, what happpens if it shortens to **exactly** 1 hour? Then the finishing constraint is redundant.)

7 Duality

7.1 Pricing interpretation

Back to our usual manufacturing LP problem. For the sake of illustration, we drop the 3rd constraint, and consider the items as *blocks of wood* and *cans of paint* (instead of shop hours).

Manufacturer

$$\begin{array}{rcrcrl} \text{Max} & 3x_1 & + & 2x_2 & & \\ & x_1 & + & x_2 & \leq 80 & \text{[wood]} \\ & 2x_1 & + & x_2 & \leq 100 & \text{[paint]} \\ & & & x_1, x_2 & \geq 0 & \end{array}$$

Market

Prices:
$y_1 =$ price (in \$) of one block of wood
$y_2 =$ price (in \$) of one can of paint

Manufacturer owns 80 blocks of wood and 100 cans of paint. He can sell his stock at market prices or buy additional stock at market prices. He can also produce and sell goods (toys) using the available stock.

What is his best strategy (assuming everything produced will be sold)?

- ⋆ Selling stock generates a profit of $80y_1 + 100y_2$.
- ⋆ If the cost (in market prices) of producing x_1 *toy soldiers* is strictly less than the sale price, i.e. if
$$y_1 + 2y_2 < 3$$
then there is **no limit** on the profit of manufacturer. He can generate arbitrarily large profit by buying additional stock to produce toy soldiers in arbitrary amounts.

  Why? The manufacturer can produce x_1 toy soldiers by purchasing x_1 blocks of wood, and $2x_1$ additional cans of paint. He pays $x_1(y_1 + 2y_2)$ and makes $3x_1$ in sales. Net profit is then $x_1(3 - y_1 - 2y_2)$. Now, if $y_1 + 2y_2 < 3$, say if $y_1 + 2y_2 \leq 2.9$, then the net profit is then $x_1(3 - y_1 - 2y_2) \geq (3 - 2.9) = 0.1x_1$. So making arbitrarily many x_1 toy soldiers generates a profit of $0.1x_1$ (arbitrarily high).
- ⋆ Similarly, **no limit** on the profit if the cost of producing x_2 *toy trains* is less than the sale price, i.e. if
$$y_1 + y_2 < 2$$
- ⋆ Market prices are **non-negative**.

Market (the competition) will not allow the manufacturer to make arbitrarily large profit. It will set its prices so that the manufacturer makes as little as possible. The market is thus solving the following:

$$\left.\begin{array}{rcrcrl} \text{Min} & 80y_1 & + & 100y_2 & & \\ & y_1 & + & 2y_2 & \geq 3 & \text{[toy soldiers]} \\ & y_1 & + & y_2 & \geq 2 & \text{[toy trains]} \\ & & & y_1, y_2 & \geq 0 & \end{array}\right\} \textbf{Dual} \text{ of the manufacturing problem}$$

Estimating the optimal value

$$\begin{array}{rrrrrl} \text{Max} & 3x_1 & + & 2x_2 & & \\ & x_1 & + & x_2 & \leq 80 & \text{[wood]} \\ & 2x_1 & + & x_2 & \leq 100 & \text{[paint]} \\ & & & x_1, x_2 & \geq 0 & \end{array}$$

Before solving the LP, the manufacturer wishes to get a quick rough estimate (upper bound) on the value of the optimal solution. For instance, the objective function is $3x_1 + 2x_2$ which is certainly less than $3x_1 + \mathbf{3}x_2$, since the variables x_1, x_2 are non-negative. We can rewrite this as $3(x_1 + x_2)$ and we notice that $x_1 + x_2 \leq 80$ by the first constraint. Together we have:

$$z \;=\; 3x_1 + 2x_2 \;\leq\; 3x_1 + 3x_2 \;\leq\; 3(x_1 + x_2) \;\leq\; 3 \times 80 \;=\; \$240$$

Conclusion is that every production plan will generate no more than \$240, i.e., the value of any feasible solution (including the optimal one) is not more than 240. Likewise we can write:

$$z \;=\; 3x_1 + 2x_2 \;\leq\; 4x_1 + 2x_2 \;\leq\; 2(2x_1 + x_2) \;\leq\; 2 \times 100 \;=\; \$200$$

since $2x_1 + x_2 \leq 100$ by the 2nd constraint. We can also combine constraints for an even better estimate:

$$z \;=\; 3x_1 + 2x_2 \;\leq\; (x_1 + x_2) + (2x_1 + x_2) \;\leq\; 80 \;+\; 100 \;=\; \$180$$

In general, we consider $y_1 \geq 0$, $y_2 \geq 0$ and take y_1 times the 1st contraint + y_2 times the 2nd constraint.

$$y_1(x_1 + x_2) + y_2(2x_1 + x_2) \;\leq\; 80y_1 \;+\; 100y_2$$

We can rewrite this expression by collecting coefficients of x_1 and x_2:

$$(y_1 + 2y_2)x_1 + (y_1 + y_2)x_2 \;\leq\; 80y_1 \;+\; 100y_2$$

In this expression, if the **coefficient** of x_1 is **at least** 3 and the coefficient of x_2 is **at least** 2, i.e., if

$$y_1 + 2y_2 \geq 3$$
$$y_1 + y_2 \geq 2$$

then, just like before, we obtain an upper bound on the value of $z = 3x_1 + 2x_2$:

$$z \;=\; 3x_1 + 2x_2 \;\leq\; (y_1 + 2y_2)x_1 + (y_1 + y_2)x_2 \;=\; y_1(x_1 + x_2) + y_2(2x_1 + x_2) \;\leq\; 80y_1 \;+\; 100y_2$$

If we want the best possible upper bound, we want this expression be as small as possible.

$$\left.\begin{array}{rrrrrl} \text{Min} & 80y_1 & + & 100y_2 & & \\ & y_1 & + & 2y_2 & \geq 3 & \\ & y_1 & + & y_2 & \geq 2 & \\ & & & y_1, y_2 & \geq 0 & \end{array}\right\} \text{The } \textbf{Dual} \text{ problem}$$

The original problem is then called the **Primal** problem.

Primal

$$\begin{array}{rrrrrl} \text{Max} & 3x_1 & + & 2x_2 & & \\ & x_1 & + & x_2 & \leq 80 & \\ & 2x_1 & + & x_2 & \leq 100 & \\ & & & x_1, x_2 & \geq 0 & \end{array}$$

$$\begin{array}{rrrrrl} \text{Min} & 80y_1 & + & 100y_2 & & \\ & y_1 & + & 2y_2 & \geq 3 & \\ & y_1 & + & y_2 & \geq 2 & \\ & & & y_1, y_2 & \geq 0 & \end{array}$$

Dual

Matrix formulation

In general, for maximization problem with $\leq$ inequalities, the dual is obtained simply by

- transposing (flipping around the diagonal) the matrix $\mathbf{A}$,
- swapping vectors $\mathbf{b}$ and $\mathbf{c}$,
- switching the inequalities to $\geq$, and
- changing max to min.

$$\begin{array}{rr} \max \mathbf{c}^T\mathbf{x} & \min \mathbf{b}^T\mathbf{y} \\ \mathbf{A}\mathbf{x} \leq \mathbf{b} & \mathbf{A}^T\mathbf{y} \geq \mathbf{c} \\ \mathbf{x} \geq 0 & \mathbf{y} \geq 0 \end{array}$$

7.2 Duality Theorems and Feasibility

Theorem 3 (Weak Duality Theorem)**.** *If* $\mathbf{x}$ *is any feasible solution of the primal and* $\mathbf{y}$ *is any feasible solution of the dual, then*

$$\mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}$$

In other words, the value of **any** feasible solution to the **dual** yields an upper bound on the value of any feasible solution (including the optimal) to the **primal**.

$$\mathbf{c}^T\mathbf{x} \leq (\mathbf{A}^T\mathbf{y})^T\mathbf{x} = (\mathbf{y}^T\mathbf{A})\mathbf{x} = \mathbf{y}^T(\mathbf{A}\mathbf{x}) \leq \mathbf{y}^T\mathbf{b} = \mathbf{b}^T\mathbf{y}$$

Consequently, if **primal** is *unbounded*, then **dual** must be *infeasible* and likewise, if **dual** is *unbounded*, then **primal** must be *infeasible*. Note that is it possible that both **primal** and **dual** are *infeasible*. But if both are *feasible*, then neither of them is *unbounded*.

dual \ **primal**	infeasible	feasible bounded	unbounded
infeasible	✓	✗	✓
feasible bounded	✗	✓	✗
unbounded	✓	✗	✗

✓ possible
✗ impossible

Strong Duality and Complementary Slackness

Theorem 4 (Strong Duality Theorem)**.** *If* $\mathbf{x}$ *is an* **optimal** *solution to the primal and* $\mathbf{y}$ *is an* **optimal** *solution to the dual, then*

$$\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$$

Moreover, $\mathbf{y}^T(\underbrace{\mathbf{b}-\mathbf{A}\mathbf{x}}_{\text{slack in primal}}) = 0$ *and* $\mathbf{x}^T(\underbrace{\mathbf{A}^T\mathbf{y}-\mathbf{c}}_{\text{slack in dual}}) = 0.$

In simple terms: whenever a constraint is **not tight** (has a positive slack) in the **primal**, then the **dual** variable corresponding to this constraint must be 0. Conversely, if a **primal** variable is strictly positive, then the corresponding **dual** constraint must be tight (slack is zero).

This can be seen as follows (note that $\mathbf{x} \geq 0$, $\mathbf{y} \geq 0$, $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ and $\mathbf{A}^T\mathbf{y} \geq \mathbf{c}$)

$$0 \leq \mathbf{y}^T(\mathbf{b}-\mathbf{A}\mathbf{x}) = \mathbf{y}^T\mathbf{b} - \mathbf{y}^T\mathbf{A}\mathbf{x} = \mathbf{b}^T\mathbf{y} - (\mathbf{A}^T\mathbf{y})^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y} - \mathbf{c}^T\mathbf{x} = 0$$
$$0 \leq \mathbf{x}^T(\mathbf{A}^T\mathbf{y}-\mathbf{c}) = (\mathbf{y}^T\mathbf{A}-\mathbf{c}^T)\mathbf{x} = \mathbf{y}^T(\mathbf{A}\mathbf{x}) - \mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{b} - \mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y} - \mathbf{c}^T\mathbf{x} = 0$$

7.3 General LPs

If LP contains equalities or unrestricted variables, these can be also handled with ease. In particular,

equality constraint corresponds to an **unrestricted** variable, and **vice-versa**.

Why? Notice that when we produced an upper bound, we considered only non-negative $y_1 \geq 0$, since multiplying the $\leq$ constraint by a negative value changes the sign to $\geq$ and thus the upper bound becomes a lower bound instead. However, if the constraint was an equality (i.e. if we had $x_1 + x_2 = 80$ instead), we could allow negative y_1 as well and still produce an upper bound. For instance, we could write

$$3x_1 + 2x_2 \leq 5x_1 + 2x_2 = (-1) \times (\underbrace{x_1 + x_2}_{=80}) + 3 \times (\underbrace{2x_1 + x_2}_{\leq 100}) \leq -80 + 3 \times 100 = \$220$$

So we would make y_1 unrestricted.

Conversely, if some variable in our problem, say x_1, were unrestricted in sign (could be negative as well), then we could **not** conclude that $3x_1 + 2x_2 \leq 4x_1 + 2x_2$ holds for all feasible solutions, as we did in our 2nd estimate; namely if x_1 is **negative**, then this is **not true** (it is actually $>$ rather than $\leq$). However, if x_1 is unrestricted but $x_2 \geq 0$, we could still conclude that $3x_1 + 2x_2 \leq 3x_1 + 2x_2$, since the coefficient of x_1 is not changing in this expression. In our general expression, we had $(y_1 + 2y_2)x_1$ and we demanded that the coefficient $y_1 + 2y_2$ of x_1 is at least 3 for the

upper bound to work. If x_1 is **unrestricted**, we can simply insist that the coefficient $y_1 + 2y_2$ **equals** 3 to make the upper bound work.

The same way we can conclude that

$\geq$ constraint corresponds to an **non-positive** variable, and **vice-versa**.

Primal (Max)	Dual (Min)
i-th constraint $\leq$	variable $y_i \geq 0$
i-th constraint $\geq$	variable $y_i \leq 0$
i-th constraint $=$	variable y_i unrestricted
$x_i \geq 0$	i-th constraint $\geq$
$x_i \leq 0$	i-th constraint $\leq$
x_i unrestricted	i-th constraint $=$

$$\begin{aligned} \text{Max } & 3x_1 + 2x_2 + x_3 \\ & x_1 + x_2 + \tfrac{1}{2}x_3 \leq 80 \\ & 2x_1 + x_2 + x_3 = 100 \\ & x_1 + x_3 \geq 40 \\ & x_1 \text{ unrestricted} \\ & x_2 \leq 0 \\ & x_3 \geq 0 \end{aligned}$$

Primal

$$\begin{aligned} \text{Min } & 80y_1 + 100x_2 + 40x_3 \\ & y_1 + 2y_2 + y_3 = 3 \\ & y_1 + y_2 \leq 2 \\ & \tfrac{1}{2}y_1 + y_2 + y_3 \geq 1 \\ & y_1 \geq 0 \\ & y_2 \text{ unrestricted} \\ & y_3 \leq 0 \end{aligned}$$

Dual

7.4 Complementary slackness

$$\begin{aligned} \max\ & 6x_1 + x_2 - x_3 - x_4 \\ & x_1 + 2x_2 + x_3 + x_4 \leq 5 \\ & 3x_1 + x_2 - x_3 \leq 8 \\ & x_2 + x_3 + x_4 = 1 \\ & x_2, x_3, x_4 \geq 0 \\ & x_1 \text{ unrestricted} \end{aligned}$$

We wish to check if one of the following assignments is an optimal solution.

a) $x_1 = 2, x_2 = 1, x_3 = 0, x_4 = 0$
b) $x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0$

To this end, we use **Complementary Slackness**. Let us discuss the theory first.

Theory

As usual, let $\mathbf{x}$ denote the vector of variables, let $\mathbf{c}$ be the vector of coefficients of variables of the objective function, let $\mathbf{A}$ be the coefficient matrix of the left-hand side of our constraints, and let $\mathbf{b}$ be the vector of the right-hand side of the constraints. Let $\mathbf{y}$ be the variables of the dual.

$$\begin{aligned} \max\ & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \quad \text{PRIMAL} \\ & \mathbf{x} \geq 0 \end{aligned} \qquad \begin{aligned} \min\ & \mathbf{b}^{\mathrm{T}}\mathbf{y} \\ & \mathbf{A}^{\mathrm{T}}\mathbf{y} \geq \mathbf{c} \quad \text{DUAL} \\ & \mathbf{y} \geq 0 \end{aligned}$$

We say that vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_m)$ are **complementary** if

$$\mathbf{y}^T(\underbrace{\mathbf{b} - \mathbf{A}\mathbf{x}}_{\substack{\text{slack in} \\ \text{primal}}}) = 0 \text{ and } \mathbf{x}^T(\underbrace{\mathbf{A}^T\mathbf{y} - \mathbf{c}}_{\substack{\text{slack in} \\ \text{dual}}}) = 0$$

In other words,

- whenever $y_i > 0$, then **x** satisfies the i-th constraint with equality ("the constraint is **tight**")
- whenever $x_i > 0$, then **y** satisfies the i-th constraint of the dual with equality

Exercise. Show that the shadow prices π defined by (a basic solution) **x** are always complementary to **x**.

Recall that **Strong Duality** this says that if **x** is an optimal solution to the primal and **y** is an optimal solution to the dual, then $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$. In fact, more is true.

Complementary Slackness (and some consequences): Assume that **x** is an optimal solution to the primal.

- If **y** is an **optimal** solution to the dual, then **x** and **y** are **complementary**.
- If **y** is a **feasible** solution in the dual and is **complementary** to **x**, then **y** is **optimal** in the dual.
- There **exists** a **feasible** solution **y** to the dual such that **x** and **y** are **complementary**.

Notice that the last bullet follows from our observation about shadow prices. Another consequence of this is:

If **x** is a basic feasible primal solution and π are the corresponding shadow prices, then **x** is **optimal** if and only if π is a **feasible** solution of the dual

If we have equalities, $\geq$-inequalities, unrestricted or non-positive variables, everything works just the same.

Back to example

To check if the provided solutions are optimal, we need the dual.

$$\begin{array}{rrrrrrrl}
\max & 6x_1 & + & x_2 & - & x_3 & - & x_4 \\
 & x_1 & + & 2x_2 & + & x_3 & + & x_4 \leq 5 \\
 & 3x_1 & + & x_2 & - & x_3 & & \leq 8 \\
 & & & x_2 & + & x_3 & + & x_4 = 1 \\
 & & & & & & & x_2, x_3, x_4 \geq 0 \\
 & & & & & & & x_1 \text{ unrestricted}
\end{array}$$

$$\begin{array}{rrrrrrrl}
\min & 5y_1 & + & 8y_2 & + & y_3 & & \\
 & y_1 & + & 3y_2 & & & = & 6 \\
 & 2y_1 & + & y_2 & + & y_3 & \geq & 1 \\
 & y_1 & - & y_2 & + & y_3 & \geq & -1 \quad \text{DUAL} \\
 & y_1 & & & + & y_3 & \geq & -1 \\
 & & & & & y_1, y_2 & \geq & 0 \\
 & & & & & y_3 \text{ unrestricted} & &
\end{array}$$

a) $x_1 = 2, x_2 = 1, x_3 = 0, x_4 = 0$ $\rightarrow$ assume **x** is optimal, then

$\rightarrow$ there are y_1, y_2, y_3 such that $\mathbf{y} = (y_1, y_2, y_3)$ is feasible in the dual and complementary to **x**

check 1st primal constraint: $x_1 + 2x_2 + x_3 + x_4 = 2 + 2 + 0 + 0 = 4 < 5$ **not tight**

$\rightarrow$ therefore y_1 must be 0 because **y** is complementary to **x**

check 2nd primal constraint: $3x_1 + x_2 - x_3 = 6 + 1 - 0 = 7 < 8$ **not tight**

$\rightarrow$ therefore y_2 must be 0 because **y** is complementary to **x**

Knowing this, check the 1st dual constraint: $y_1 + 3y_2 = 0 + 0 = 0 \neq 6$

$\rightarrow$ this shows that (y_1, y_2, y_3) **not feasible** in the dual, but we assume that it is.

This means that our assumptions were wrong and so (x_1, x_2, x_3, x_4) is **not optimal**.

b) $x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0$ $\rightarrow$ again assume that **x** is optimal, then

$\rightarrow$ there are y_1, y_2, y_3 such that $\mathbf{y} = (y_1, y_2, y_3)$ is feasible in the dual and complementary to **x**

check 1st primal constraint: $x_1 + 2x_2 + x_3 + x_4 = 3 + 0 + 1 + 0 = 4 < 5$ **not tight** $\rightarrow$ $y_1 = 0$

check 2nd primal constraint: $3x_1 + x_2 - x_3 = 9 + 0 - 1 = 8$ **tight**

check 3rd primal constraint: $x_1 + x_2 + x_3 = 1$ **tight**

check sign constraints: $x_2, x_3, x_4 \geq 0 \rightarrow$ we conclude that (x_1, x_2, x_3, x_4) is **feasible** in the primal

Now we look at values in **x** with respect to the dual constraints:

x_1 is unrestricted $\rightarrow$ 1st dual constraint $y_1 + 3y_2 = 6$ is (always) **tight**

$x_3 > 0$ we deduce $\rightarrow$ 3rd dual constraint must be **tight**: $y_1 - y_2 + y_3 = -1$

Together we have

$$\begin{array}{rcl} y_1 & = & 0 \\ y_1 + 3y_2 & = & 6 \\ y_1 - y_2 + y_3 & = & -1 \end{array}$$

This has a unique solution $y_1 = 0, y_2 = 2, y_3 = 1$. By construction, this solution is **complementary** to **x**.

The last step is to **check** if **y** is also **feasible** in the dual. We already checked 1st and 3rd dual constraint.

check 2nd dual constraint: $2y_1 + y_2 + y_3 = 0 + 2 + 1 = 3 \geq 1 \rightarrow$ the constraint is **satisfied**

check 4th dual constraint: $y_1 + y_3 = 0 + 1 \geq -1$ the constraint is **satisfied**

check sign restrictions: $y_1 = 0 \geq 0$ and $y_2 = 2 \geq 0 \rightarrow$ sign restrictions **satisfied**

$\rightarrow$ this shows that (y_1, y_2, y_3) indeed a **feasible** solution to the dual.

From this we can conclude that (x_1, x_2, x_3, x_4) is indeed **optimal**.

Summary

- given **x**, check if **x** is feasible
- then find which variables y_i should be 0
- then find which dual constraints should be tight
- this yields a system of equations
- solve the system
- verify that the solution is feasible in the dual

If all these steps succeed, then the given **x** is indeed optimal; otherwise, it is not.

Question: what happens if **x** is feasible but not a basic solution ?

Review

$$\begin{array}{rcl} \text{Max } 3x_1 + 2x_2 & & \\ x_1 + x_2 & \leq & 80 \\ 2x_1 + x_2 & \leq & 100 \\ x_1 & \leq & 40 \\ x_1, x_2 & \geq & 0 \end{array}$$

Original problem

$$\begin{array}{rcl} x_3 & = & 80 - x_1 - x_2 \\ x_4 & = & 100 - 2x_1 - x_2 \\ x_5 & = & 40 - x_1 \\ \hline z & = & 0 + 3x_1 + 2x_2 \end{array}$$

Initial dictionary

$$\begin{array}{rcl} x_1 & = & 20 + x_3 - x_4 \\ x_2 & = & 60 - 2x_3 + x_4 \\ x_5 & = & 20 - x_3 + x_4 \\ \hline z & = & 180 - x_3 - x_4 \end{array}$$

Optimal dictionary

Add a new activity: *toy cars*, $\frac{1}{2}$h carving, 1h finishing, 1 unit towards demand limit, \$1 price $\rightarrow x_6 =$ #cars

$$\begin{array}{rcl} \text{Max } 3x_1 + 2x_2 + x_6 & & \\ x_1 + x_2 + \frac{1}{2}x_6 & \leq & 80 \\ 2x_1 + x_2 + x_6 & \leq & 100 \\ x_1 + x_6 & \leq & 40 \\ x_1, x_2, x_6 & \geq & 0 \end{array}$$

Original problem

$$\begin{array}{rcl} x_3 & = & 80 - x_1 - x_2 - \frac{1}{2}x_6 \\ x_4 & = & 100 - 2x_1 - x_2 - x_6 \\ x_5 & = & 40 - x_1 - x_6 \\ \hline z & = & 0 + 3x_1 + 2x_2 + x_6 \end{array}$$

Initial dictionary

?

Optimal dictionary

in the optimal dictionary $\rightarrow$ make x_6 *non-basic* (no production of cars) $\rightarrow$ **feasible** modified dictionary

$$\begin{array}{rcl} x_3 + \frac{1}{2}x_6 & = & 80 - x_1 - x_2 \\ x_4 + x_6 & = & 100 - 2x_1 - x_2 \\ x_5 + x_6 & = & 40 - x_1 \\ \hline z - x_6 & = & 0 + 3x_1 + 2x_2 \end{array}$$

$$\begin{array}{rcl} x_3' & = & 80 - x_1 - x_2 \\ x_4' & = & 100 - 2x_1 - x_2 \\ x_5' & = & 40 - x_1 \\ \hline z' & = & 0 + 3x_1 + 2x_2 \end{array}$$

$\rightarrow$

$$\begin{array}{rcl} x_1 & = & 20 + x_3' - x_4' \\ x_2 & = & 60 - 2x_3' + x_4' \\ x_5' & = & 20 - x_3' + x_4' \\ \hline z' & = & 180 - x_3' - x_4' \end{array}$$

substitute: $x_3' = x_3 + \frac{1}{2}x_6, \quad x_4' = x_4 + x_6, \quad x_5' = x_5 + x_6, \quad z' = z - x_6$

$$\begin{array}{rcl} x_1 &=& 20 + (x_3 + \frac{1}{2}x_6) - (x_4 + x_6) \\ x_2 &=& 60 - 2(x_3 + \frac{1}{2}x_6) + (x_4 + x_6) \\ x_5 + x_6 &=& 20 - (x_3 + \frac{1}{2}x_6) + (x_4 + x_6) \\ \hline z - x_6 &=& 180 - (x_3 + \frac{1}{2}x_6) - (x_4 + x_6) \end{array} \qquad \begin{array}{rcl} x_1 &=& 20 + x_3 - x_4 - \frac{1}{2}x_6 \\ x_2 &=& 60 - 2x_3 + x_4 \\ x_5 &=& 20 - x_3 + x_4 - \frac{1}{2}x_6 \\ \hline z &=& 180 - x_3 - x_4 - \frac{1}{2}x_6 \end{array}$$

Add a new constraint: *packaging*, 250 units of cardboard, 3/soldier, 4/train, 1/car $\rightarrow x_7 =$ slack

$$\begin{array}{rcl} \text{Max } 3x_1 + 2x_2 + x_6 & & \\ x_1 + x_2 + \frac{1}{2}x_6 &\leq& 80 \\ 2x_1 + x_2 + x_6 &\leq& 100 \\ x_1 + x_6 &\leq& 40 \\ 3x_1 + 4x_2 + x_6 &\leq& 250 \\ x_1, x_2, x_6 &\geq& 0 \end{array} \qquad \begin{array}{rcl} x_3 &=& 80 - x_1 - x_2 - \frac{1}{2}x_6 \\ x_4 &=& 100 - 2x_1 - x_2 - x_6 \\ x_5 &=& 40 - x_1 - x_6 \\ x_7 &=& 250 - 3x_1 - 4x_2 - x_6 \\ \hline z &=& 0 + 3x_1 + 2x_2 + x_6 \end{array}$$

in the optimal dictionary $\rightarrow$ make x_7 **basic** $\rightarrow$ expres x_7 using the dictionary $\rightarrow$ new dictionary

$x_7 = 250 - 3x_1 - 4x_2 - x_6 = 250 - 3(20 + x_3 - x_4 - \frac{1}{2}x_6) - 4(60 - 2x_3 + x_4) - x_6 = -50 + 5x_3 - x_4 + \frac{1}{2}x_6$

$$\begin{array}{rcl} x_1 &=& 20 + x_3 - x_4 - \frac{1}{2}x_6 \\ x_2 &=& 60 - 2x_3 + x_4 \\ x_5 &=& 20 - x_3 + x_4 - \frac{1}{2}x_6 \\ \hline z &=& 180 - x_3 - x_4 - \frac{1}{2}x_6 \end{array} \quad \rightarrow \quad \begin{array}{rcl} x_1 &=& 20 + x_3 - x_4 - \frac{1}{2}x_6 \\ x_2 &=& 60 - 2x_3 + x_4 \\ x_5 &=& 20 - x_3 + x_4 - \frac{1}{2}x_6 \\ x_7 &=& -50 + 5x_3 - x_4 + \frac{1}{2}x_6 \\ \hline z &=& 180 - x_3 - x_4 - \frac{1}{2}x_6 \end{array}$$

If the resulting dictionary is feasible, then it is also optimal (we don't change z, all coeffs still non-positive)

However, the resulting dictionary may be **infeasible** if some basic variable is negative (here $x_7 < 0$)

$\rightarrow$ to recover optimal solution, we use **Dual Simplex Method**.

8 Other Simplex Methods

8.1 Dual Simplex Method

- we use when the dictionary is **infeasible** but **dually feasible**
- same as Simplex on the Dual **without** dealing with the Dual directly
- useful when adding new constraints to quickly recover optimal solution

Recall: for every (basic) solution of the Primal, we have **shadow prices** that we can assign to each item (constraint) in such a way that the total value of items in shadow prices is **exactly** the value of the solution

PRIMAL

$$\begin{aligned} \text{Max } & 3x_1 + 2x_2 \\ & x_1 + x_2 \le 80 \\ & 2x_1 + x_2 \le 100 \\ & x_1 \le 40 \\ & x_1, x_2 \ge 0 \end{aligned}$$

$$\begin{aligned} x_1 &= 40 - x_5 \\ x_3 &= 40 - x_2 + x_5 \\ x_4 &= 20 - x_2 + 2x_5 \\ \hline z &= 120 + 2x_2 - 3x_5 \end{aligned}$$

Solution: $x_1 = 40$, $x_2 = 0$, $x_3 = 40$, $x_4 = 20$, $x_5 = 0$ of value $z = 120$

Shadow prices: $\pi_1 = 0$, $\pi_2 = 0$, $\pi_3 = 3$

$80\pi_1 + 100\pi_2 + 40\pi_3 = 120 = 3x_1 + 2x_3$

DUAL

$$\begin{aligned} \text{Min } & 80y_1 + 100y_2 + 40y_3 \\ & y_1 + 2y_2 + y_3 \ge 3 \\ & y_1 + y_2 \ge 2 \\ & y_1, y_2, y_3 \ge 0 \end{aligned}$$

Note: the above shadow prices $\rightarrow$ an **infeasible** solution in Dual

$$\begin{aligned} 80\pi_1 + 100\pi_2 + 40\pi_3 &= 120 \quad \text{the same value as Primal} \\ \pi_1 + 2\pi_2 + \pi_3 &= 3 \quad \ge 3 \\ \pi_1 + \pi_2 &= 0 \quad \not\ge 2 \end{aligned}$$

$\pi_1 = 0 \ge 0$, $\pi_2 = 0 \ge 0$, $\pi_3 = 3 \ge 0$

Shadow prices corresponding to a non-optimal **feasible** solution of the Primal are **infeasible** in the Dual
Shadow prices corresponding to an **optimal** solution of the Primal are **optimal** in the Dual.
Shadow prices for an **infeasible** solution of the Primal **may or may not** be **feasible** in the Dual.

A solution of the Primal is **dually feasible** if the corresponding shadow prices are **feasible** in the dual.

Dual Simplex Algorithm

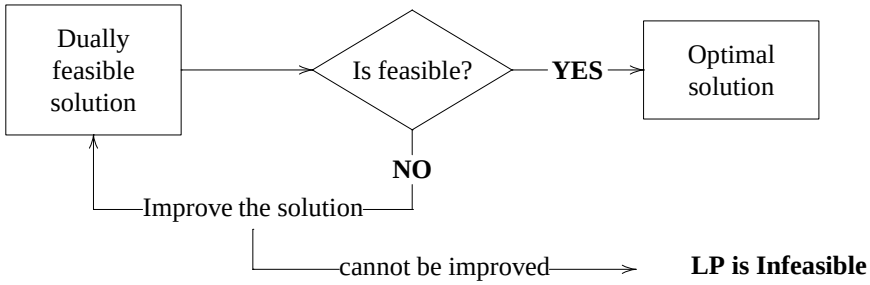
We **maintain** that the current solution is **dually feasible** but may itself be **infeasible**

$$\begin{aligned} x_1 &= 20 + x_3 - x_4 - \tfrac{1}{2}x_6 \\ x_2 &= 60 - 2x_3 + x_4 \\ x_5 &= 20 - x_3 + x_4 - \tfrac{1}{2}x_6 \\ x_7 &= -50 + 5x_3 - x_4 + \tfrac{1}{2}x_6 \\ \hline z &= 180 - x_3 - x_4 - \tfrac{1}{2}x_6 \end{aligned}$$

$\leftarrow$ is dually feasible

How do we know that the solution is **dually feasible**?
... if all coefficients of variables in z are **non-positive**

(Why? Because the coeffs of slack variables are shadow prices while the coeffs of non-slack variables are reduced costs)



Since $x_7 < 0$, the solution is **infeasible**. We want to make it **feasible**. In order to do that, we need to **increase** the value of x_7. We make x_7 non-basic (value $= 0$).

We can do this by increasing one of the **non-basic** variables (x_3 or x_4 or x_6), i.e., by making one of the non-basic variables **basic**. Clearly, x_4 is not good for this since increasing x_4 makes x_7 only **more** negative (x_4 appears in the equation for x_7 with negative coefficient).

So we must increase x_3 or x_6. But we also must make sure that this results in **dually feasible** dictionary, i.e., the resulting coefficients of non-basic variables in z are non-positive.

If x_6 enters the basis, then $x_7 = -50 + 5x_3 - x_4 + \frac{1}{2}x_6 \quad \rightarrow \quad \frac{1}{2}x_6 = 50 - 5x_3 + x_4 + x_7$ and so

$$z = 180 - x_3 - x_4 - \tfrac{1}{2}x_6 = 180 - x_3 - x_4 - \tfrac{1}{2} \times 2(50 - 5x_3 + x_4 + x_7) = 130 + 4x_3 - 2x_4 - x_7$$

If x_3 enters the basis, then $x_7 = -50 + 5x_3 - x_4 + \frac{1}{2}x_6 \quad \rightarrow \quad 5x_3 = 50 + x_4 - \frac{1}{2}x_6 + x_7$ and so

$$z = 180 - x_3 - x_4 - \tfrac{1}{2}x_6 = 180 - 1 \times \tfrac{1}{5}(50 + x_4 - \tfrac{1}{2}x_6 + x_7) - x_4 - \tfrac{1}{2}x_6 = 170 - \tfrac{6}{5}x_4 - \tfrac{2}{5}x_6 - \tfrac{1}{5}x_7$$

When x_6 enters the basis, the solution is **not** dually feasible, while it **is** when x_3 enters the basis. How do we find out? Both substitutions are a result of **adding** to z some multiple Δ of the equation for x_7.

$$z = 180 - x_3 - x_4 - \tfrac{1}{2}x_6 - \Delta(\overbrace{50 - 5x_3 + x_4 - \tfrac{1}{2}x_6 + x_7}^{0})$$

$$= (180 - 50\Delta) + (\underbrace{-1 + 5\Delta}_{\leq 0})x_3 + (\underbrace{-1 - \Delta}_{\leq 0})x_4 + (\underbrace{-\tfrac{1}{2} + \tfrac{1}{2}\Delta}_{\leq 0})x_6 + (\underbrace{-\Delta}_{\leq 0})x_7$$

$$\begin{array}{l} -1 + 5\Delta \leq 0 \\ -1 - \Delta \leq 0 \\ -\frac{1}{2} + \frac{1}{2}\Delta \leq 0 \\ -\Delta \leq 0 \end{array} \quad \rightarrow \quad \begin{array}{l} \Delta \leq \frac{1}{5} \\ -1 \leq \Delta \\ \Delta \leq 1 \\ 0 \leq \Delta \end{array} \quad \rightarrow \quad \boxed{0 \leq \Delta \leq \tfrac{1}{5}} \quad \rightarrow \quad \begin{array}{l} \text{for } \Delta = \frac{1}{5} \text{ the bound is tight for } x_3 \\ \rightarrow x_3 \text{ disappears from } z \rightarrow \text{enters the basis} \end{array}$$

Ratio test

Simplified procedure: compare coefficients of non-basic variables in z and in x_7, choose smallest ratio

x_3 : coeff $-1x_3$ in z and $5x_3$ in x_7, ratio $\dfrac{1}{5} = 0.2$

x_4 : coeff $-1x_4$ in z and $-1x_4$ in x_7, *no constraint* (negative coeff in x_7)

x_6 : coeff $-\frac{1}{2}x_6$ in z and $\frac{1}{2}x_6$ in x_7, ratio $\dfrac{\frac{1}{2}}{\frac{1}{2}} = 1$

$$\begin{array}{rcrcrcrcr} x_1 & = & 20 & + & x_3 & - & x_4 & - & \frac{1}{2}x_6 \\ x_2 & = & 60 & - & 2x_3 & + & x_4 & & \\ x_5 & = & 20 & - & x_3 & + & x_4 & - & \frac{1}{2}x_6 \\ x_7 & = & -50 & + & 5x_3 & - & x_4 & + & \frac{1}{2}x_6 \\ \hline z & = & 180 & - & 1x_3 & - & x_4 & - & \frac{1}{2}x_6 \end{array}$$

ratio for x_3: $\dfrac{1}{5} = 0.2$

(again **watch-out:** we only consider this ratio because the coefficient of x_3 is positive)

Smallest ratio $\frac{1}{5}$ for $x_3 \to x_3$ enters

$$x_7 = -50 + 5x_3 - x_4 + \tfrac{1}{2}x_6$$
$$\to \quad x_3 = 10 + \tfrac{1}{5}x_4 - \tfrac{1}{10}x_6 + \tfrac{1}{5}x_7$$

$$\begin{aligned} x_1 &= 30 - \tfrac{4}{5}x_4 - \tfrac{3}{5}x_6 + \tfrac{1}{5}x_7 \\ x_2 &= 40 + \tfrac{3}{5}x_4 + \tfrac{1}{5}x_6 - \tfrac{2}{5}x_7 \\ x_3 &= 10 + \tfrac{1}{5}x_4 - \tfrac{1}{10}x_6 + \tfrac{1}{5}x_7 \\ x_5 &= 10 + \tfrac{4}{5}x_4 - \tfrac{2}{5}x_6 - \tfrac{1}{5}x_7 \\ \hline z &= 170 - \tfrac{6}{5}x_4 - \tfrac{2}{5}x_6 - \tfrac{1}{5}x_7 \end{aligned}$$

Shadow prices:
$\pi_1 = 0$, $\pi_2 = \frac{6}{5}$,
$\pi_3 = 0$, $\pi_4 = \frac{1}{5}$
optimal solution to the Dual.
(Why?)

Solution is feasible (and dually feasible) $\to$ **optimal** solution found.

Summary

Starting with a **dually feasible** solution:

1. Find a **basic** variable x_i of negative value.
2. If no such x_i exists $\to$ stop, the solution is feasible $\to$ **optimal** solution found.
3. **Ratio test**: in the dictionary, in the equation for x_i, find a non-basic variable x_j such that
   - x_j appears with **positive** coefficient a in the equation for x_i
   - the ratio $\dfrac{c}{a}$ is smallest possible (where $-c$ is the coefficient of x_j in z)
4. If no such x_j exists $\to$ stop, no feasible solution exists $\to$ report that LP is **Infeasible**.
5. **Pivot** x_j into the basis, x_i leaves the basis.
   (the resulting dictionary is **guaranteed** to be dually feasible)
6. Repeat.

8.2 Upper-Bounded Simplex

Back to the toy factory problem. Subsequent market survey revealed the maximum demand for trains to be 50.

$$\begin{aligned} \text{Max } & 3x_1 + 2x_2 \\ & x_1 + x_2 \le 80 \\ & 2x_1 + x_2 \le 100 \\ & x_1 \le 40 \\ & x_2 \le 50 \\ & x_1, x_2 \ge 0 \end{aligned}$$

$$\begin{aligned} \text{Max } & 3x_1 + 2x_2 \\ & x_1 + x_2 \le 80 \\ & 2x_1 + x_2 \le 100 \\ & 0 \le x_1 \le 40 \\ & 0 \le x_2 \le 50 \end{aligned}$$

introduce new (complementary) variables x_1', x_2':

$$\begin{aligned} x_1' &= 40 - x_1 \\ x_2' &= 50 - x_2 \\ 0 &\le x_1' \le 40 \\ 0 &\le x_2' \le 50 \end{aligned}$$

We solve the problem as usual, but for each variable we keep track of either the variable or its complement but never both. That is, the dictionary will either contain x_1 or x_1' but not both. Note that it is not necessary to keep both as one can be derived from the other by a simple substitution $x_1' = 40 - x_1$. In a way, we may think of having both x_1 and x_1' in the dictionary but one of them is **hidden**.

We modify the **ratio test** to account for these hidden variables. For illustration, let us solve the above.

$$\begin{aligned} x_3 &= 80 - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned}$$
$$0 \le x_3 \le \infty$$
$$0 \le x_4 \le \infty$$

x_1 increases to $\Delta \ge 0$ (enters)
variables change as follows:

$$\begin{aligned} x_1 &= \Delta & 0 &\le x_1 \le 40 \\ x_3 &= 80 - \Delta & 0 &\le x_3 \le \infty \\ x_4 &= 100 - 2\Delta & 0 &\le x_4 \le \infty \end{aligned}$$

$$\begin{aligned} 0 &\le \Delta \le 40 \\ 0 &\le 80 - \Delta \le \infty \\ 0 &\le 100 - 2\Delta \le \infty \end{aligned}$$

$$\begin{aligned} \Delta &\le 40 \\ \Delta &\le \tfrac{80}{1} = 80 \\ \Delta &\le \tfrac{100}{2} = 50 \end{aligned}$$

Most restrictive constrained imposed by the upper bound on $x_1 \to$ we replace x_1 by $x_1' = 40 - x_1$.

$$\begin{aligned} x_3 &= 80 - x_1 - x_2 \\ x_4 &= 100 - 2x_1 - x_2 \\ \hline z &= 0 + 3x_1 + 2x_2 \end{aligned}
\quad \to \quad
\begin{aligned} x_3 &= 80 - (40 - x_1') - x_2 \\ x_4 &= 100 - 2(40 - x_1') - x_2 \\ \hline z &= 0 + 3(40 - x_1') + 2x_2 \end{aligned}
\quad \to \quad
\begin{aligned} x_3 &= 40 + x_1' - x_2 \\ x_4 &= 20 + 2x_1' - x_2 \\ \hline z &= 120 - 3x_1' + 2x_2 \end{aligned}$$

Ratio test

three possible types of constraints

(1) *lower bound* on a basic variables (x_3, x_4) $\rightarrow$ usual ratio test, var with a **negative** coeff of $x_1 \rightarrow$ pivot
(2) *upper bound* on the entering variable (x_1) $\rightarrow$ replace the variable by its complementary ($'$) variable
(3) *upper bound* on a basic variable (see below) $\rightarrow$ var with a **positive** coeff of $x_1 \rightarrow$ replace by $'$ var

$$\begin{array}{rcrcrcr} x_3 &=& 40 &+& x_1' &-& x_2 \\ x_4 &=& 20 &+& 2x_1' &-& x_2 \\ \hline z &=& 120 &-& 3x_1' &+& 2x_2 \end{array}$$

x_2 increased to Δ, ratio test:
(1) $x_3 = 40 - \Delta \geq 0$
$x_4 = 20 - \Delta \geq 0$
(2) $x_2 = \Delta \leq 50$
(3) no constraint

$x_3 : \Delta \leq \frac{40}{1} = 40$
$x_4 : \Delta \leq \frac{20}{1} = 20$
$x_2 : \Delta \leq 50$
x_4 leaves, we pivot

$$x_4 = 20 + 2x_1' - x_2$$
$$\downarrow$$
$$x_2 = 20 + 2x_1' - x_4$$

$$\begin{array}{rcrcrcr} x_2 &=& & & (20 + 2x_1' - x_4) & & \\ x_3 &=& 40 &+& x_1' &-& (20 + 2x_1' - x_4) \\ \hline z &=& 120 &-& 3x_1' &+& 2(20 + 2x_1' - x_4) \end{array} \quad \rightarrow \quad \begin{array}{rcrcrcr} x_2 &=& 20 &+& 2x_1' &-& x_4 \\ x_3 &=& 20 &-& x_1' &+& x_4 \\ \hline z &=& 160 &+& x_1' &-& 2x_4 \end{array}$$

x_1' increased, ratio test:
(1) $x_3 = 20 - \Delta \geq 0$
(2) $x_1' = \Delta \leq 40$
(3) $x_2 = 20 + 2\Delta \leq 50$

$x_3 : \Delta \leq \frac{20}{1} = 20$
$x_1' : \Delta \leq 40$
$x_2 : \Delta \leq \frac{50-20}{2} = 15$

Most restrictive constraint imposed by x_2
$\rightarrow$ replace by $x_2' = 50 - x_2$

$$\begin{array}{rcrcrcr} (50 - x_2') &=& 20 &+& 2x_1' &-& x_4 \\ x_3 &=& 20 &-& x_1' &+& x_4 \\ \hline z &=& 160 &+& x_1' &-& 2x_4 \end{array} \quad \rightarrow \quad \begin{array}{rcrcrcr} x_2' &=& 30 &-& 2x_1' &+& x_4 \\ x_3 &=& 20 &-& x_1' &+& x_4 \\ \hline z &=& 160 &+& x_1' &-& 2x_4 \end{array}$$

x_1' increased, ratio test:
(1) $x_2' : \frac{30}{2} = 15$
$x_3 : \frac{20}{1} = 20$
(2) $x_1' : 40$
(3) no constraint

x_2' leaves, x_1' enters
$x_2' = 30 - 2x_1' + x_4$
$2x_1' = 30 - x_2' + x_4$
$x_1' = 15 - \frac{1}{2}x_2' + \frac{1}{2}x_4$

$$\begin{array}{rcl} x_1' &=& (15 - \frac{1}{2}x_2' + \frac{1}{2}x_4) \\ x_3 &=& 20 - (15 - \frac{1}{2}x_2' + \frac{1}{2}x_4) + x_4 \\ \hline z &=& 160 + (15 - \frac{1}{2}x_2' + \frac{1}{2}x_4) - 2x_4 \end{array}$$

$$\begin{array}{rcrcrcr} x_1' &=& 15 &-& \frac{1}{2}x_2' &+& \frac{1}{2}x_4 \\ x_3 &=& 5 &+& \frac{1}{2}x_2' &+& \frac{1}{2}x_4 \\ \hline z &=& 175 &-& \frac{1}{2}x_2' &-& \frac{3}{2}x_4 \end{array}$$

optimal solution found
$\rightarrow$ basic variables: $x_1' = 15$, $x_3 = 5$
$\rightarrow$ non-basic variables: $x_2' = 0$, $x_4 = 0$

$x_1 = 40 - x_1' = 25$
$x_2 = 50 - x_2' = 50$
$x_3 = 5$
$x_4 = 0$

Note: Be careful when expressing the values of variables from the dictionary not to confuse the variables and their complements. In the above, x_2' is **non-basic** and so its value is **zero**. However, this does not mean that x_2 is and in fact, the value of x_2 is $50 - x_2' = 50$. In particular, observe that both x_2 and x_2' can **never** be non-basic at the same time (one of them is always positive). The fact that we don't see a variable in the final dictionary does not mean its value is zero.

8.3 Lower bounds

The above procedure allows us to also handle general lower bounds (other than 0).

If a variable x_i is constrained by bounds $\ell_i \leq x_i \leq r_i$ where ℓ_i, r_i are numbers, then we introduce a new variable x_j and substitute $x_i = (x_j + \ell_i)$. In the resulting problem, the variable x_j is non-negative with an upper bound $x_j \leq r_i - \ell_i$. (If $r_i = \infty$, we don't have an upper bound for x_j.) We then solve the problem using the new variable, and then calculate x_i from the resulting value of x_j by taking $x_i = x_j + \ell_i$.

Contractual obligations with a big retailer demand that at least 30 toy soldiers be produced.

$$\begin{array}{rrrl} \text{Max} & 3x_1 & + 2x_2 & \\ & x_1 & + x_2 & \leq 80 \\ & 2x_1 & + x_2 & \leq 100 \\ & 30 \leq x_1 & \leq 40 & \\ & 0 \leq x_2 & \leq 50 & \end{array}$$

introduce x_5 to replace x_1:

$$x_1 = 30 + x_5$$
$$0 \leq x_5 \leq 10$$

$$\begin{array}{rrrl} \text{Max} & 3(30+x_5) & + 2x_2 & \\ & (30+x_5) & + x_2 & \leq 80 \\ & 2(30+x_5) & + x_2 & \leq 100 \\ & 0 \leq x_5 & \leq 10 & \\ & 0 \leq x_2 & \leq 50 & \end{array}$$

$$\begin{array}{rrrl} \text{Max} & 2x_2 & + 3x_5 & + 90 \\ & x_2 & + x_5 & \leq 50 \\ & x_2 & + 2x_5 & \leq 40 \\ & 0 \leq x_5 & \leq 10 & \\ & 0 \leq x_2 & \leq 50 & \end{array}$$

introduce complementary variables x_2', x_5':

$$x_2' = 50 - x_2$$
$$x_5' = 10 - x_5$$
$$0 \leq x_2' \leq 50$$
$$0 \leq x_5' \leq 10$$

Initial dictionary:

$$\begin{array}{rrrrr} x_3 & = 50 & - x_2 & - x_5 & \\ x_4 & = 40 & - x_2 & - 2x_5 & \\ \hline z & = 90 & + 2x_2 & + 3x_5 & \end{array}$$

$\rightarrow$ x_5 enters, ratio test:

(1) $x_3 : \frac{50}{1} = 50$

(2) $x_4 : \frac{40}{2} = 20$

(3) $x_5 : 10$

(4) no constraint

$\rightarrow$ x_5 replaced by $x_5 = 10 - x_5'$

$$\begin{array}{rrrrr} x_3 & = 40 & - x_2 & + x_5' & \\ x_4 & = 20 & - x_2 & + 2x_5' & \\ \hline z & = 120 & + 2x_2 & - 3x_5' & \end{array}$$

$\rightarrow$ x_2 enters, ratio test:

(1) $x_3 : \frac{40}{1} = 40$

(2) $x_4 : \frac{20}{1} = 20$

(3) $x_2 : 50$

(4) no constraint

$\rightarrow$ x_4 leaves, $x_2 = 20 - x_4 + 2x_5'$

$$\begin{array}{rrrrr} x_2 & = 20 & - x_4 & + 2x_5' & \\ x_3 & = 20 & + x_4 & - x_5' & \\ \hline z & = 160 & - 2x_4 & + x_5' & \end{array}$$

$\rightarrow$ x_5' enters, ratio test

(1) $x_2 : \frac{20}{1} = 20$

(2) $x_5' : 10$

(3) $x_2 : \frac{50-20}{2} = 15$

$\rightarrow$ x_5' replaced by $x_5' = 10 - x_5$

$$\begin{array}{rrrrr} x_2 & = 40 & - x_4 & - 2x_5 & \\ x_3 & = 10 & + x_4 & + x_5 & \\ \hline z & = 170 & - 2x_4 & - x_5 & \end{array}$$

optimal solution:

$x_2 = 40$, $x_3 = 10$, $x_4 = 0$, $x_5 = 0$ $\rightarrow$ $x_1 = 30$, $x_2 = 40$ solution of value $z = 170$

calculate x_1 from $x_5 \rightarrow x_1 = 30 + x_5 = 30$

8.4 Dual Simplex with Upper Bounds

Instead of selling via a retailer, we decided to ship directly to customers. This involves extra packaging step before shipping requiring 3 sheets of cardboard per a toy soldier, and 4 sheets per a toy train.

Being a green company, we wish to minimize the amount of packaging material used. However, we also want to make profit, namely at least \$150. This leads to the following minimization problem.

$$\begin{array}{rrrll} \text{Min} & 3x_1 & + 4x_2 & & \\ & x_1 & + x_2 & \leq 80 & \text{[carving]} \\ & 2x_1 & + x_2 & \leq 100 & \text{[finishing]} \\ & 3x_1 & + 2x_2 & \geq 150 & \text{[profit]} \\ & 0 \leq x_1 & \leq 40 & & \\ & 0 \leq x_2 & \leq 50 & & \end{array}$$

introduce x_1', x_2' and slack/excess x_3, x_4, x_5

$$x_1' = 40 - x_1$$
$$x_2' = 50 - x_1$$
$$0 \leq x_1' \leq 40$$
$$0 \leq x_2' \leq 50$$

$$0 \leq x_3 \leq \infty$$
$$0 \leq x_4 \leq \infty$$
$$0 \leq x_5 \leq \infty$$

$$\text{Max } z = -3x_1 - 4x_2$$

$$\begin{array}{rrrrr} x_3 & = 80 & - x_1 & - x_2 & \\ x_4 & = 100 & - 2x_1 & - x_2 & \\ x_5 & = -150 & + 3x_1 & + 2x_2 & \\ \hline z & = 0 & - 3x_1 & - 4x_2 & \end{array}$$

dictionary is **dually feasible** $\rightarrow$ all coefficients in z are non-positive $\rightarrow$ can use Dual Simplex method

first, we check **upper bounds** of basic variables: $x_3 = 80 \leq \infty$, $x_4 = 100 \leq \infty$, $x_5 = -150 \leq \infty \rightarrow$ OK

next, we check **lower bounds**: $x_3 = 80 \geq 0$, $x_4 = 100 \geq 0$, $x_5 = -150 \not\geq 0 \rightarrow$ not OK, must increase x_5

x_5 leaves, **ratio test** $\rightarrow x_1 : \dfrac{3}{3} = 1$, $x_2 : \dfrac{4}{2} = 2$ (compare coefficients in z and (positive coeffs) in x_5)

take smallest $\rightarrow x_1$ enters, $x_5 = -150 + 3x_1 + 2x_2$ $\rightarrow$ $x_1 = 50 - \frac{2}{3}x_2 + \frac{1}{3}x_5$

$$\begin{array}{rcl} x_1 &=& (50 - \frac{2}{3}x_2 + \frac{1}{3}x_5) \\ x_3 &=& 80 - (50 - \frac{2}{3}x_2 + \frac{1}{3}x_5) - x_2 \\ x_4 &=& 100 - 2(50 - \frac{2}{3}x_2 + \frac{1}{3}x_5) - x_2 \\ \hline z &=& 0 - 3(50 - \frac{2}{3}x_2 + \frac{1}{3}x_5) - 4x_2 \end{array} \quad \rightarrow \quad \begin{array}{rcl} x_1 &=& 50 - \frac{2}{3}x_2 + \frac{1}{3}x_5 \\ x_3 &=& 30 - \frac{1}{3}x_2 - \frac{1}{3}x_5 \\ x_4 &=& \frac{1}{3}x_2 - \frac{2}{3}x_5 \\ \hline z &=& -150 - 2x_2 - x_5 \end{array}$$

check **upper bounds**: $x_1 = 50 \not\leq 40$, $x_3 = 30 \leq \infty$, $x_4 = 0 \leq \infty \rightarrow$ not OK, replace x_1 by $x_1 = 40 - x_1'$

$$\begin{array}{rcl} (40 - x_1') &=& 50 - \frac{2}{3}x_2 + \frac{1}{3}x_5 \\ x_3 &=& 30 - \frac{1}{3}x_2 - \frac{1}{3}x_5 \\ x_4 &=& \frac{1}{3}x_2 - \frac{2}{3}x_5 \\ \hline z &=& -150 - 2x_2 - x_5 \end{array} \quad \rightarrow \quad \begin{array}{rcl} x_1' &=& -10 + \frac{2}{3}x_2 - \frac{1}{3}x_5 \\ x_3 &=& 30 - \frac{1}{3}x_2 - \frac{1}{3}x_5 \\ x_4 &=& \frac{1}{3}x_2 - \frac{2}{3}x_5 \\ \hline z &=& -150 - 2x_2 - x_5 \end{array}$$

upper bounds: $x_1' = -10 \leq 40$, $x_3 = 30 \leq \infty$, $x_4 = 0 \leq \infty \rightarrow$ OK

lower bounds: $x_1' = -10 \not\geq 0$, $x_3 = 30 \geq 0$, $x_4 = 0 \geq 0 \rightarrow$ not OK, must increase x_1'

x_1' leaves, ratio test $\rightarrow x_2 = \dfrac{2}{(2/3)} = 3$, x_5 : *no constraint* (negative coefficient in x_1') $\rightarrow x_2$ enters

$x_1' = -10 + \frac{2}{3}x_2 - \frac{1}{3}x_5$

$\frac{2}{3}x_2 = 10 + x_1' + \frac{1}{3}x_5$

$\rightarrow x_2 = 15 + \frac{3}{2}x_1' + \frac{1}{2}x_5$

$$\begin{array}{rcl} x_2 &=& (15 + \frac{3}{2}x_1' + \frac{1}{2}x_5) \\ x_3 &=& 30 - \frac{1}{3}(15 + \frac{3}{2}x_1' + \frac{1}{2}x_5) - \frac{1}{3}x_5 \\ x_4 &=& \frac{1}{3}(15 + \frac{3}{2}x_1' + \frac{1}{2}x_5) - \frac{2}{3}x_5 \\ \hline z &=& -150 - 2(15 + \frac{3}{2}x_1' + \frac{1}{2}x_5) - x_5 \end{array} \quad \begin{array}{rcl} x_2 &=& 15 + \frac{3}{2}x_1' + \frac{1}{2}x_5 \\ x_3 &=& 25 - \frac{1}{2}x_1' - \frac{1}{2}x_5 \\ x_4 &=& 5 + \frac{1}{2}x_1' - \frac{1}{2}x_5 \\ \hline z &=& -180 - 3x_1' - 2x_5 \end{array}$$

upper bounds: $x_2 = 15 \leq 50$, $x_3 = 25 \leq \infty$, $x_4 = 5 \leq \infty \rightarrow$ OK

lower bounds: $x_2 = 15 \not\geq 0$, $x_3 = 25 \geq 0$, $x_4 = 5 \geq 0 \rightarrow$ OK

$\Big\} \Longrightarrow$ **optimal solution** found

values of variables: basic $x_2 = 15$, $x_3 = 25$, $x_4 = 5$, non-basic $x_5 = 0$, $x_1' = 0 \quad \rightarrow \quad x_1 = 40 - x_1' = 40$

$\rightarrow x_1 = 40$, $x_2 = 15$ is optimal with value $z = -180$

$\Longrightarrow$ Minimum amount of packaging required to make \$150 of profit is 180 units.

8.5 Goal Programming

We consider again the (usual) toy factory production problem with two products, *toy soldiers* and *toy trains*, with sale prices \$3 respectively \$2, requiring 1 hour each in the carving shop, and 2 hours respectively 1 hour in the finishing shop; the total hours being 80 and 100 respectively, as usual. Each toy soldier requires 3 units of packaging material, and each toy train requires 4.

The company became non-profit and set itself several new goals.

Goal #1: not to produce more than 40 toy soldiers; only up to 40 can be sold at the price \$3

Goal #2: make at least \$140 of profit to pay workers wages

Goal #3: use at most 130 units of packaging material; only this much is currently available

Each goal can be met individually, however trying to meet all three at once may not be (and it is not) possible. *Failure* to meet a goal carries a **penalty**, proportional to the amount by which the goal is not met.

Penalty #1: each toy soldier over 40 units will not be sold; penalty of \$3 of missing profit

Penalty #2: any profit below \$140 has to be obtained by borrowing at 20% interest

Penalty #3: any missing packaging material has to be bought at \$0.80 a unit

Our combined **goal** is to **minimize** the **total penalty** incured. This leads to the following:

$$
\begin{array}{rcrl}
x_1 + x_2 & \leq & 80 & \text{[Carving]} \\
2x_1 + x_2 & \leq & 100 & \text{[Finishing]} \\
x_1 & \leq & 40 & \text{[Goal \#1]} \\
3x_1 + 2x_2 & \geq & 140 & \text{[Goal \#2]} \\
3x_1 + 4x_2 & \leq & 130 & \text{[Goal \#3]}
\end{array}
\quad \xrightarrow[slack]{add} \quad
\begin{array}{rcl}
x_1 + x_2 + x_3 & = & 80 \\
2x_1 + x_2 + x_4 & = & 100 \\
x_1 + x_5 & = & 40 \\
3x_1 + 2x_2 - x_6 & = & 140 \\
3x_1 + 4x_2 + x_7 & = & 130
\end{array}
$$

Slack variables x_3 and x_4 are assumed to be always non-negative; the come from **hard** constraints – must be satisfied **unconditionally**. On the other hand, we can allow x_5, x_6, and x_7 to be unrestricted, since the corresponding constraints are **soft** – we may violate any of them, but we incur a penalty.

Depending on whether each of x_5, x_6, x_7 is positive or negative, we incur different penalties. For instance, if x_5 is positive, there is no penalty, the penalty is 0; if x_5 is negative, the penalty is $-3x_5$, since we lose \$3 for each unsold toy soldier (there is "minus" in penalty because x_5 is negative, not because we lose profit).

Similarly, if x_6 is positive, the penalty is 0, while if x_6 is negative, the penalty is $-1.2x_6$, since if profit drops below \$140, to pay the workers we have borrow at 20% interest. Note that we can specify different penalties for both positive and negative slack (not necessarily only zero) and the method works the same way.

The question now is: **how do we formulate this problem as a linear program?**

We express each of the variables x_5, x_6, x_7 as a **difference** of two **non-negative** variables. Namely we write x_5 as $x_5 = x_5^+ - x_5^-$ where x_5^+ and x_5^- are new non-negative variables. Similary we write $x_6 = x_6^+ - x_6^-$ and $x_7 = x_7^+ - x_7^-$. As we discussed at the beginning of the course, under some assumptions, the values of these new variables (in an optimal solution) will be precisely the positive/negative parts of the variables x_5, x_6, x_7.

This allow us to **construct** an objective function that captures the sum of all the penalties that we incur in different situations (depending on the signs of the variables x_5, x_6, x_7).

$$
\begin{array}{l}
x_5 = x_5^+ - x_5^- \\
x_6 = x_6^+ - x_6^- \\
x_7 = x_7^+ - x_7^- \\
x_5^+, x_5^-, x_6^+, x_6^-, x_7^+, x_7^- \geq 0
\end{array}
\qquad
\begin{array}{ll}
\min & 3x_5^- + 1.2x_6^- + 0.8x_7^- \\
& \begin{array}{rcl}
x_1 + x_2 + x_3 & = & 80 \\
2x_1 + x_2 + x_4 & = & 100 \\
x_1 + x_5^+ - x_5^- & = & 40 \\
3x_1 + 2x_2 - x_6^+ + x_6^- & = & 140 \\
3x_1 + 4x_2 + x_7^+ - x_7^- & = & 130
\end{array}
\end{array}
$$

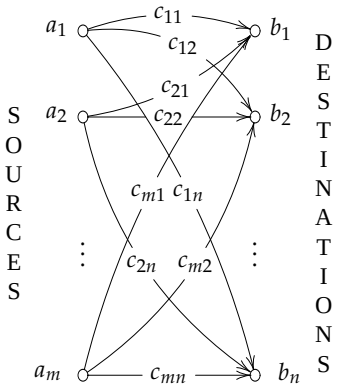
We solve this problem using standard methods (Simplex method or Dual Simplex method). Alternatively, we can solve the dual and from that construct a solution to this problem.

Exercise. Construct the dual LP to the above problem.

9 Transportation Problem

Historical note

The problem and its solution described first in 1941 by Hitchcock – predates the Simplex Method – independently discovered by Koopmans during World War II used to minimize shipping times for cargo ships – the method spearheaded research on linear and non-linear programs in economics. The Assignment problem (a special case of the Transportation Problem) can be solved more efficiently using the Hungarian Method, found in the work of Egerváry in 1931 and first explicitely described in 1955 by Kuhn who named it the Hungarian Method; recently it was found that the method was actually first found a century earlier by Jacobi in the context of solving systems of ordinary differential equations (work published posthumously in 1890). Since then, more efficient (but more complicated) algorithms have been found for the two problems.



In the transportation problem, we have

- *m sources* (warehouses, factories) producing items, and
- *n destinations* (shops, businesses) requiring these items.

The items need to be **transported** from sources to destinations which has **associated cost**.

The goal is to **minimize** the total **cost** of **transportation**.

- The i-th source has a_i **available** items
- The j-th destination **demands** b_j items to be delivered.
- It costs c_{ij} to deliver one item from i-th source to j-th destination.

Assumption: destinations do not care from which source the items come, and sources do not care to which destinations they deliver.

Decision variables: x_{ij} = the number of items transported from the i-th source to j-th destination

Balanced Transportation Problem

$$\text{Minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\begin{aligned}
\text{subject to} \quad \sum_{j=1}^{n} x_{ij} &= a_i \quad \text{for } i = 1,\dots,m && [i\text{-th sources has } a_i \text{ available items}]\\
\sum_{i=1}^{m} x_{ij} &= b_i \quad \text{for } j = 1,\dots,n && [j\text{-th demands } b_i \text{ items}]\\
x_{ij} &\geq 0 \quad \text{for } i = 1,\dots,m \text{ and } j = 1,\dots,n && [\text{non-negative amounts transported}]
\end{aligned}$$

A solution exists if and only if $\overbrace{\sum_{i=1}^{m} a_i}^{\text{total supply}} = \overbrace{\sum_{i=1}^{n} b_i}^{\text{total demand}}$ (the total supply is equal to the total demand)

Why equalities? If the total supply is bigger than the total demand by $\Delta > 0$, then we introduce a *dummy* destination with demand Δ and with **zero cost** of transportation from all sources. If the total demand is bigger than the total supply by $\Delta > 0$, then **no feasible solution** exists. However, we can still model the **shortage** by introducing a *dummy* source with supply Δ and some the costs of transportation (say zero).

Example

A furniture company owns three warehouses in the New York City area and needs to deliver chairs to its three shops in the city for tomorrow. The three shops demand 4, 2, and 3 units respectively. Current stock level of chairs in the three warehouses is 8, 6, and 3 respectively. Delivery costs from each warehouse to each store are different due to different distances. These are as follows (in $ per unit).

		Shop #		
Delivery costs:	Warehouse #1	\$7	\$3	\$4
	Warehouse #2	\$4	\$2	\$2
	Warehouse #3	\$2	\$1	\$5

Demand:
Shop #1: 4 units
Shop #2: 2 units
Shop #4: 3 units

Supply:
Warehouse #1: 8 units
Warehouse #2: 6 units
Warehouse #3: 3 units

Find the least expensive way to deliver the chairs to the stores.

Total Demand: 9 units
Total Supply: 17 units

} Solution exists, since total supply is at least the total demand. The excess supply of 8 units will be assigned to a **dummy destination** at zero cost.

		Shops #			
Delivery costs:	Warehouse #1	\$7	\$3	\$4	\$0
	Warehouse #2	\$4	\$2	\$2	\$0
	Warehouse #3	\$2	\$1	\$5	\$0

Demand:
Shop #1: 4 units
Shop #2: 2 units
Shop #4: 3 units
Dummy: 8 units

Supply:
Warehouse #1: 8 units
Warehouse #2: 6 units
Warehouse #3: 3 units

$$\text{Min } 7x_{11} + 3x_{12} + 4x_{13} + 0x_{14} + 4x_{21} + 2x_{22} + 2x_{23} + 0x_{24} + 2x_{31} + 1x_{32} + 5x_{33} + 0x_{34}$$

$$\begin{aligned}
\text{s.t. } & x_{11} + x_{12} + x_{13} + x_{14} = 8 \\
& x_{21} + x_{22} + x_{23} + x_{24} = 6 \\
& x_{31} + x_{32} + x_{33} + x_{34} = 3 \\
& x_{11} + x_{21} + x_{31} = 4 \\
& x_{12} + x_{22} + x_{32} = 2 \\
& x_{13} + x_{23} + x_{33} = 3 \\
& x_{14} + x_{24} + x_{34} = 8
\end{aligned}$$

all variables non-negative

9.1 Transportation Simplex Method

Tableau form:

c_{11} x_{11}	c_{12} x_{12}	c_{13} x_{13}	c_{14} x_{14}	u_1 a_1
c_{21} x_{21}	c_{22} x_{22}	c_{23} x_{23}	c_{24} x_{24}	u_2 a_2
c_{31} x_{31}	c_{32} x_{32}	c_{33} x_{33}	c_{34} x_{34}	u_3 a_3
v_1 b_1	v_2 b_2	v_3 b_3	v_4 b_4	$z =?$

7	3	4	0	u_1 8
4	2	2	0	u_2 6
2	1	5	0	u_3 3
v_1 4	v_2 2	v_3 3	v_4 8	

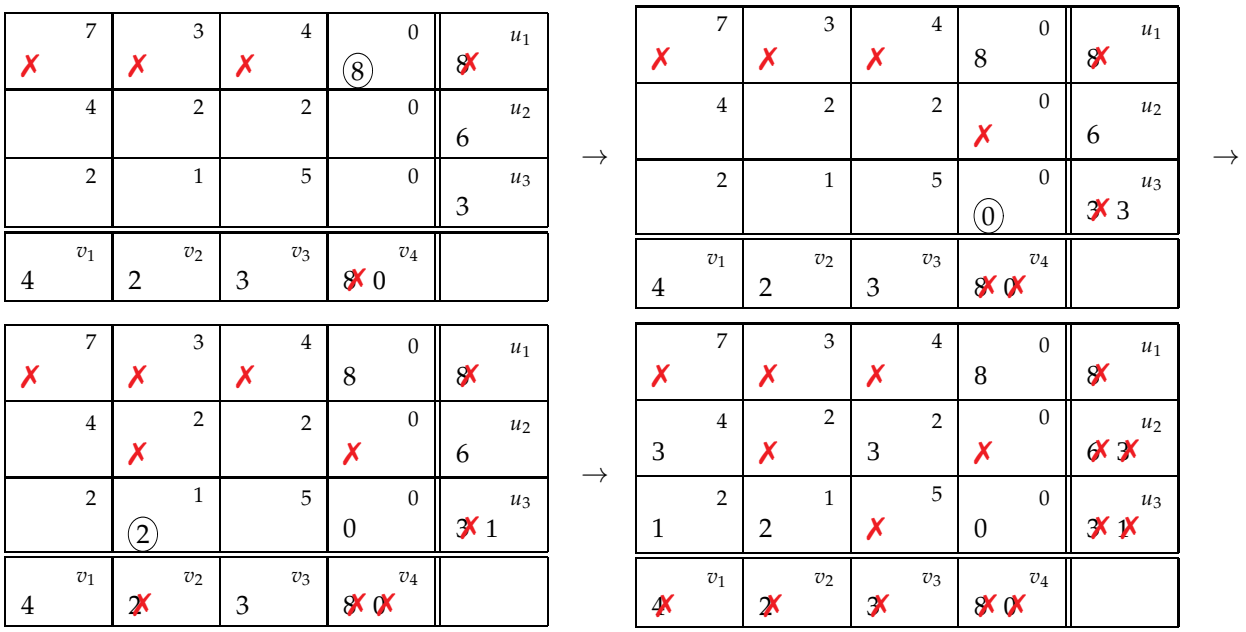
Shadow prices

- u_1, u_2, u_3 for the first three constraints (*supply constraints*),
- v_1, v_2, v_3, v_4 for the latter four constraints (*demand constraints*)

Initialization

We need to find a starting **basic feasible solution**. One method is the **Minimum-cost** rule:

1. If the tableau has only 1 row, then for every variable x_{ij}: set $x_{ij} = b_j$, and add x_{ij} to the basis; then stop.
2. Otherwise find a variable x_{ij} of smallest cost c_{ij}.
3. If $a_i \leq b_j$, then set $x_{ij} = a_i$, add x_{ij} to the basis, remove the i-th row, and decrease b_j to $b_j - a_i$.
4. If $a_i > b_j$, then set $x_{ij} = b_j$, add x_{ij} to the basis, remove the j-th column, and decrease a_i to $a_i - b_j$.
5. Repeat.



Basic Feasible Solutions

The problem with m sources and n destinations has $m \times n$ variables and $m + n$ constraint. The constraints are **not independent**, but each $m + n - 1$ of them are. This implies the **fundamental property**:

- every **basis** contains **exactly** $m + n - 1$ **variables**,
- each **constraint** contains **at least** one **basic** variable.

We **mark** all **cells** that correspond to the **basic variables**.

- the table will contain $m + n - 1$ marked cells
- every unmarked cell has value zero
- each row contains at least one marked cell
- each column contains at least one marked cell
- **consequently:** there always exists a row or column with **exactly one** marked cell.

7	3	4	0 (8)	u_1 8
4 (3)	2	2 (3)	0	u_2 6
2 (1)	1 (2)	5	0 (0)	u_3 3
v_1 4	v_2 2	v_3 3	v_4 8	$z = 22$

Shadow prices

Economic interpretation: $(-u_i)$ = price of chair in the i-th warehouse
v_j = sale price of chair in the j-th shop.

Optimality criterion: the current solution is optimal if for **every** warehouse i and **every** shop j the **cost of transportation** is **more** than (or equal to) the **difference in prices**, i.e., if $c_{ij} \geq v_j + u_i$

How to determine shadow prices?

If x_{ij} is **basic**, then we break even when transporting from i-th source to j-th destination, i.e., if $c_{ij} = v_j + u_i$ (we have no incentive *increasing* or *decreasing* the transported amount, since that does not change our profit).

This yields $m+n-1$ equations over $m+n$ variables $u_1,\dots,u_m$, $v_1,\dots,v_n$. This is an **underdetermined** system. Since we are only interested in price **differences**, we may simply assume that one of the shadow prices is zero. With that, the system will have a solution which will give us the shadow prices we are interested in. **Instead** of writing down and solving the system, we find shadow prices **directly**.

Set $u_i = 0$ for some $i \in \{1,\dots,m\}$. Then repeat the following until all shadow prives have been determined.

- Find i such that u_i has been determined $\rightarrow$ for every basic variable x_{ij}, set $v_j = c_{ij} - u_i$.
- Find j such that v_j has been determined $\rightarrow$ for every basic variable x_{ij}, set $u_i = c_{ij} - v_j$.

We choose to start by assigning $u_2 = 0$.

7	3	4	0 (8)	u_1 8
4 (3)	2	2 (3)	0	0 (6)
2 (1)	1 (2)	5	0 (0)	u_3 3
4 4	v_2 2	2 3	v_4 8	

$\rightarrow$

7	3	4	0 (8)	u_1 8
4 (3)	2	2 (3)	0	0 6
2 (1)	1 (2)	5	0 (0)	-2 3
(4) 4	v_2 2	2 3	v_4 8	

7	3	4	0 (8)	u_1 8
4 (3)	2	2 (3)	0	0 6
2 (1)	1 (2)	5	0 (0)	(−2) 3
4 4	3 2	2 3	2 8	

$\rightarrow$

7 2	3 1	4 0	0 (8)	-2 8
4 (3)	2 3	2 (3)	0 2	0 6
2 (1)	1 (2)	5 0	0 (0)	-2 3
4 4	3 2	2 3	2 8	

We **price out** non-basic variables by taking $u_i + v_j$ (the value in blue) for every non-basic x_{ij}. This value represents the **price difference** between i-th warehouse and j-th shop. If this value is **more** than the corresponding trasportation cost c_{ij}, then it pays to make x_{ij} **basic**, i.e., transport items from i-th warehouse to j-th shop, since the net profit is positive. Such variable is x_{22}, since $c_{22} = 2$ and $u_2 + v_2 = 3$.

Pivoting to a new basis

We increase x_{22} to $\Delta > 0$ and need to find out which variable leaves the basis. This is done by finding a **loop** (a cycle, circuit) in the tableau: starting from x_{22}, choosing basic variables in turn, either from the same row, or same column where we alternate between rows and columns, and coming back to x_{22}.

7	3	4	0 (8)	-2 8
$-\Delta$ 4 (3)	2 (Δ)	2 (3)	0	-4 6
2 (1) $+\Delta$	1 (2) $-\Delta$	5	0 (0)	-2 3
0 4	3 2	6 3	2 8	

We increase x_{22} to Δ. Since to total value in column 2 must be exactly 2, we decrease x_{32} from 2 to $2-\Delta$. This decreases the total value in row 3 by Δ, so we compensate by increasing x_{31} from 1 to $1+\Delta$. Then the total in column 1 increases, so we decrease x_{21} from 3 to $3-\Delta$. Finally, all row and column totals are back in order.

Largest value Δ for which all current basic variables are non-negative is $\Delta = 2$; for this value, x_{32} decreases to 0, leaves the basis, and x_{22} enters the basis.

How to find the loop? Make the entering variable basic (circle it in the tableau). Then cross out every basic variable that belongs a row or column which has no other basic variable (or has other basic variables, but they all have been crossed out). Repeat as long as possible. What remains is the desired loop.

7	3	4	0 (8) ✗	−2
4 (3)	2 (Δ)	2 (3) ✗	0	−4
2 (1)	1 (2)	5	0 (0)	−2
0	3	6	2	
4	2	3	8	

→

7	3	4	0 (8) ✗	−2
4 (3)	2 (Δ)	2 (3) ✗	0	−4
2 (1)	1 (2)	5	0 (0) ✗	−2
0	3	6	2	
4	2	3	8	

After pivoting, we obtain the following tableau. We calculate the shadow prices, and price out cells

7	3	4	0 (8)	u_1 8
4 (1)	2 (2)	2 (3)	0	u_2 6
2 (3)	1	5	0 (0)	u_3 3
v_1 4	v_2 2	v_3 3	v_4 8	$z = 20$

→

7 2	3 0	4 0	0 (8)	−2 8
4 (1)	2 (2)	2 (3)	0 2	0 6
2 (3)	1 0	5 0	0 (0)	−2 3
4 4	2 2	2 3	2 8	$z = 20$

→ x_{24} enters the basis, increased to Δ, largest $\Delta = 0$ when x_{34} leaves the basis.

7	3	4	0 (8) ✗	−2 8
4 (1) $-\Delta$	2 (2) ✗	2 (3) ✗	0 (Δ)	0 6
2 (3) $+\Delta$	1	5	0 (0) $-\Delta$	−2 3
4 4	2 2	2 3	2 8	$z = 20$

→

7 4	3 2	4 2	0 (8)	0 8
4 (1)	2 (2)	2 (3)	0 (0)	0 6
2 (3)	1 0	5 0	0 −2	−2 3
4 4	2 2	2 3	0 8	$z = 20$

Optimal solution found.

$x_{21} = 1, x_{22} = 2, x_{23} = 3, x_{31} = 3.$

Conclusion: Least costly way to deliver the chairs to stores is to deliver no chairs from warehouse #1, deliver 1, 2, and 3 chairs from warehouse #2 to shops #1, #2, and #3 respectively, and in addition deliver 3 chairs from warehouse #3 to shop #1. The total transportation cost is $20.

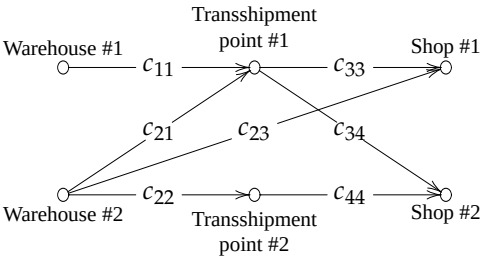
Assignment Problem

We have m machines that we want to assigned n jobs. Machines are different and to execute job j on machine i requires c_{ij} resources (time, space, energy). We want to minimize total amount of resources used.

We can model it as a Transportation Problem with m sources–machines, n destinations–jobs, costs c_{ij}, each demand a_i equal to 1, and each available supply b_j also equal 1. By adding dummy machines and jobs, we may assume that $n = m$. We may solve the problem using the standard (Simplex) method for the Transportation problem. This is unfortunately not very efficient, since **every basis** of the corresponding LP contains $n - 1$ **degenerate** basic variables. That is why specialized algorithms, such as the Hungarian Method, exist for the Assignment Problem.

Transshipment Problem

A generalization of the Transportation Problem where we may ship some items via several intermediate (**transshipment**) locations. We can model it as an ordinary Transportation Problem by **treating** each intermediate node as **both** a source and a destination (it appears twice in the tableau); its demand is equal to its supply (equal to its storage capacity), and has zero transportation cost between itself.



Transportation costs matrix:

c_{11}	∞	∞	∞	Warehouse #1
c_{21}	c_{22}	c_{23}	∞	Warehouse #2
0	∞	c_{33}	c_{34}	Transship #1
∞	0	∞	c_{44}	Transship #2
Transship #1	Transship #2	Shop #1	Shop #2	

10 Network problems

Historical note

The Shortest Path Problem is one of the most important efficient computational tools at our disposal. It is used everywhere, from GPS navigation and network communication to project management, layout design, robotics, computer graphics, and the list goes on. Often many problems reduce to finding shortest paths or use shortest paths as guidelines to optimal solutions. First algorithms for the Shortest Path problem were designed by Ford (1956), Bellman (1958), and Moore (1959). For non-negative weights, a more efficient algorithm was first suggested by Dijkstra (1959). Since then, many improved and specialized algorithms have been developed, with close to linear running time. For instance, in the context of navigation, current record holder is the Hub labelling algorithm (2011), which uses precomputed distances from carefully chosen nodes–hubs to speed up path finding on maps from hours to tenths of microseconds. For All-pairs Shortest Path problem the first algorithms were founds by Shimbel (1953), and by Roy (1959), Floyd (1962), and Warshall (1962). A more efficient approach is a combination of Dijkstra's and Bellman-Ford's algorithms known as Johnson's algorithm (1977).

The Minimum Spanning Tree problem also has a rich history. The first known algorithm was developed by Borůvka (1926) for efficient distribution of electicity. Later independently discovered by many researchers over the years: Jarník (1930), Kruskal (1956), Prim (1957), and Dijkstra (1959). All are based on similar ideas and have very efficient, almost linear-time implementations. There also exist efficient parallel and distributed implementations. The latter is notably used in minimum spanning tree protocols found frequently in routing broadcast/multicast communication in computer and wireless networking.

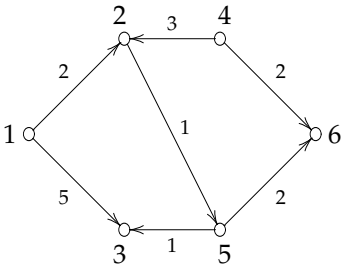
G = **graph** or **network** consists of

- a set V of **vertices** (nodes, points) and
- a set E of **edges** (arcs, lines) which are connections between vertices.

write $G = (V, E)$; write $V(G)$ for vertices of G, and $E(G)$ for edges of G.

(vertices are usually denoted u or v with subscripts; edges we usually denote e)

edges may have **direction**: an edge e between u and v may go from u to v, we write $e = (u, v)$,



vertices $V = \{1,2,3,4,5,6\}$

edges $E = \Big\{(1,2),(1,3),(2,5),(4,2),$
$(4,6),(5,3),(5,6)\Big\}$

weights $c(1,2) = 2 \quad c(1,3) = 5$
$c(2,5) = 1 \quad c(4,2) = 3 \quad c(4,6) = 2$
$c(5,3) = 1 \quad c(5,6) = 2$

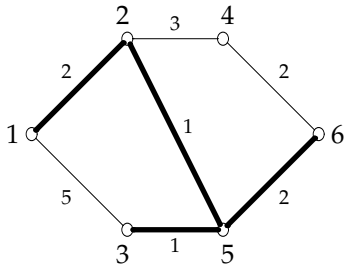


Figure 10.1: network (left), undirected network (right), edges $(1,2)$, $(2,5)$, $(3,5)$, $(5,6)$ form a tree of weight 5

or from v to u, we write $e = (v, u)$

(if an edge e does not have a direction, we treat it the same way as having both directions)

if all edges do not have a direction (are undirected), we say that the network is **undirected**

edges may have **weight**: a weight of edge $e = (u, v)$ is a real number denoted $c(e)$ or $c(u, v)$, c_e, c_{uv}

a sequence of nodes and edges $v_1, e_1, v_2, e_2, \ldots v_{k-1}, e_k, v_k$ is

- a **path** (directed path) if each e_i goes from v_i to v_{i+1}
- a **chain** (undirected path) if each e_i connects v_i and v_{i+1} (in some direction)

(often we write: $e_1, e_2, \ldots, e_k$ is a path (we omit vertices) or write: $v_1, v_2, \ldots, v_k$ is a path (we omit edges))

a network is **connected** if for every two nodes there is a path connecting them; otherwise it is **disconnected**

a **cycle** (loop, circuit) is a path starting and ending in the same node, never repeating any node or edge

a **forest** (acyclic graph) is an undirected graph that contains no cycles

a **tree** is a connected forest

Claim: A tree with n nodes contains exactly $n - 1$ edges. Adding any edge to a tree creates a cycle. Removing any edge from a tree creates a disconnected forest.

10.1 Shortest Path Problem

Given a network $G = (V, E)$ with two distinguished vertices $s, t \in V$, find a shortest path from s to t

Example: In Figure 1 (left), a shortest path from $s = 1$ to $t = 6$ is $1, 2, 5, 6$ of total length 5, while for $t = 3$ a shortest path is $1, 2, 5, 3$ of length 4. We say that **distance** from node 1 to node 6 is 5. Note that there is no path from s to $t = 4$; we indicate this by defining the distance to 4 as ∞.

LP formulation: decision variables x_{ij} for each $(i, j) \in E$

$$\text{Min} \quad \sum_{(i,j)\in E} w_{ij} x_{ij}$$

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \qquad \text{for each } i \in V$$

$$x_{ij} \in \{0, 1\} \qquad \text{for each } (i, j) \in E$$

Modeling

A new car costs \$12,000. Annual maintenance costs are as follows: $m_1 = \$2,000$ first year, $m_2 = \$4,000$ second year, $m_3 = \$5,000$ third year, $m_4 = \$9,000$ fourth year, and $m_5 = \$12,000$ fifth year and on. The car can be sold for $s_1 = \$7,000$ in the first year, for $s_2 = \$6,000$ in the second year, for $s_3 = \$2,000$ in the third year, and for $s_4 = \$1,000$ in the fourth year of ownership.

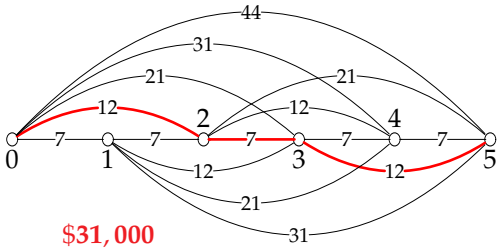
An existing car can be sold at any time and another new car purchased at \$12,000. What buying/selling strategy for the next 5 years minimizes the total cost of ownership?

Nodes = $\{0, 1, 2, 3, 4, 5\}$

Edge (i, j) represents the act of buying a car in year i and selling in year j. The weight is the price difference plus the maintanence cost, i.e., the weight is

$$c(i, j) = \$12,000 - s_{(i-j)} + m_1 + m_2 + \ldots + m_{(i-j)}$$

Answer: the length of a shortest path from node 0 to node 5.



A company wants to introduce a new model of its ever-popular *cell phone* to the market. To gauge the demand, a limited batch of product is to be brought to the market. The product is assembled from two parts, a circuit board, and housing. In addition, workers need to be trained and raw material procured.

After the initial assesment, the project manager put together a list of all tasks along with estimated duration of each task. The tasks are interdependent, some need to finish before others can start. First, (A) workers need to be trained and (B) raw materials purchased which takes 6 and 9 days, respectivelly. After these tasks are finished, both parts of the product, (C) the circuit board and (D) the housing, are manufactured taking 7 and 8 days, respectively. Each circuit board, once manufactured, needs to undergo additional testing (E) to comply with FCC regulation which takes 10 days. Afterwards, the cell phone is assembled, packaged, and shipped (F) which takes 12 days.

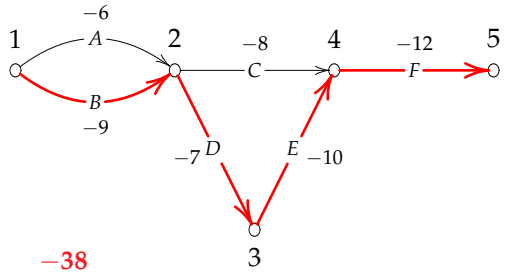
What is the minimum number of days before the product can reach the market? What is the **critical path** of the production, i.e., the sequence of tasks delaying any of which delays the whole production ?

We identify main check-points:
(1) start, (2) completion of tasks A and B,
(3) completion of task D, (4) completion of tasks C,E,
and (5) finish.

Nodes $\{1,2,3,4,5\}$

Edges correspond to tasks that connect checkpoints, weights are durations with negative sign.

Answer: A shortest path from 1 to 5 is a critical path.



(38 days = least time to hit the market)

Dijkstra's algorithm

Algorithm finds the length of a shortest path from s to every vertex of G (not only t)

Weights of edges are assumed to be **non-negative**, else the algorithm may output incorrect answer.

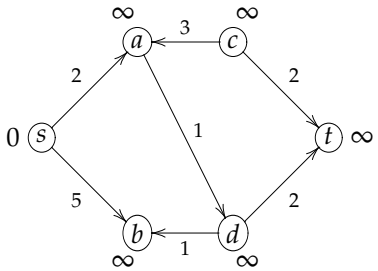
variables: d_u for each $u \in V$, an **estimate** on the distance from s to u

initialize: $d_u = \begin{cases} 0 & \text{if } u = s \\ \infty & \text{otherwise} \end{cases}$ all vertices are initially *unprocessed*

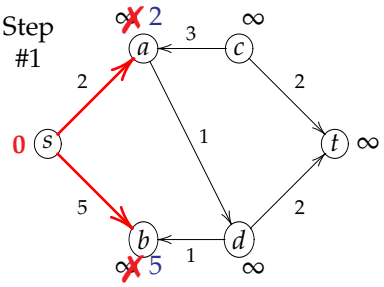
1. Find an *unprocessed* vertex u with smallest d_u
2. For each $(u,v) \in E$, update $d_v = \min\{d_v,\ d_u + c_{uv}\}$
3. Mark u as processed; repeat until all vertices are processed.
4. Report d_t as distance from s to t

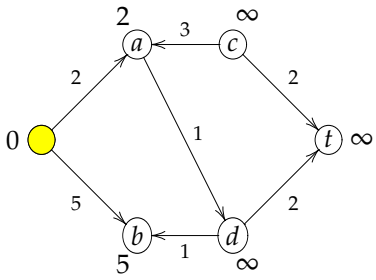
Step#	s	a	b	c	d	t
1.	$\underline{0}$	∞	∞	∞	∞	∞
		2	5			
2.	0^*	$\underline{2}$	5	∞	∞	∞
					3	
3.	0^*	2^*	5	∞	$\underline{3}$	∞
			4			5
4.	0^*	2^*	$\underline{4}$	∞	3^*	5
5.	0^*	2^*	4^*	∞	3^*	$\underline{5}$
6.	0^*	2^*	4^*	$\underline{\infty}$	3^*	5^*
final	0^*	2^*	4^*	∞^*	3^*	5^*

Example:

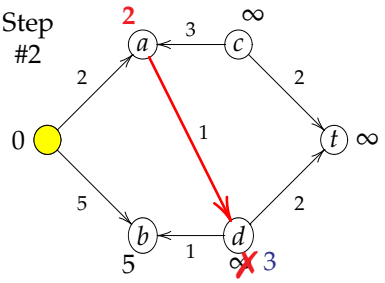


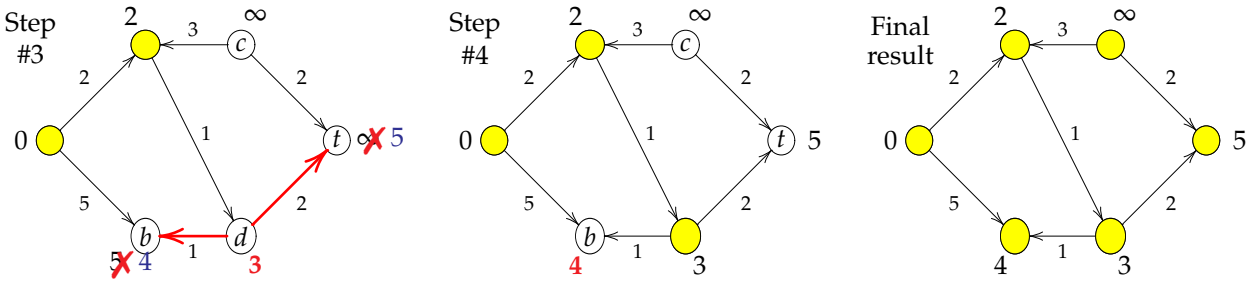
Step #1





Step #2





Can be augmented to actually find a shortest path.

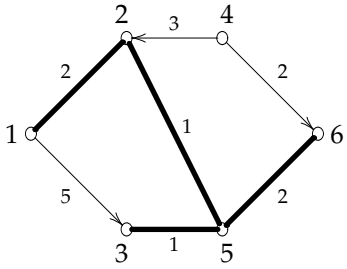
2′ For each $(u,v) \in E$, if $d_v > d_u + w_{uv}$, then $d_v = d_u + w_{uv}$ and set $p_v = u$.

For the above example, we have:

$p_1 = undefined$, $p_2 = 1$, $p_3 = 5$, $p_4 = undefined$, $p_5 = 2$, $p_6 = 5$.

We can depict this by marking the edge from p_u to u for each $u \in V$.
(the picture on right). Note that the edges we marked form a tree.

A shortest path from 1 to 6 is found backwards by taking 6, p_6, p_{p_6}, ...
The path this yields is $1, 2, 5, 6$, since $5 = p_6$, $2 = p_5$, and $1 = p_2$.



10.2 Minimum Spanning Tree

A power company delivers electricity from its power plant to neighbouring cities. The cities are interconnected by power lines operated by various operators. The power company wants to rent power lines in the grid of least total cost that will allow it to send electricity from its power plant to all cities.

Given an undirected network $G = (V, E)$ find a collection $F \subseteq E$ of minimum weight so that (V, F) is a tree.

(we say that (V, F) is a **spanning** tree because it spans all vertices)

Example: The tree in Figure 1 (right) is not spanning because it does not contain the vertex 4. Adding the edge $(2, 4)$ yields a spanning tree of weight 8, while adding the edge $(4, 6)$ yields a spanning tree of weight 7. Note that adding the edge $(1, 3)$ is not possible, for it creates a cycle $1, 2, 5, 3$ which is not allowed in a tree.

Kruskal's (Prim's) algorithm

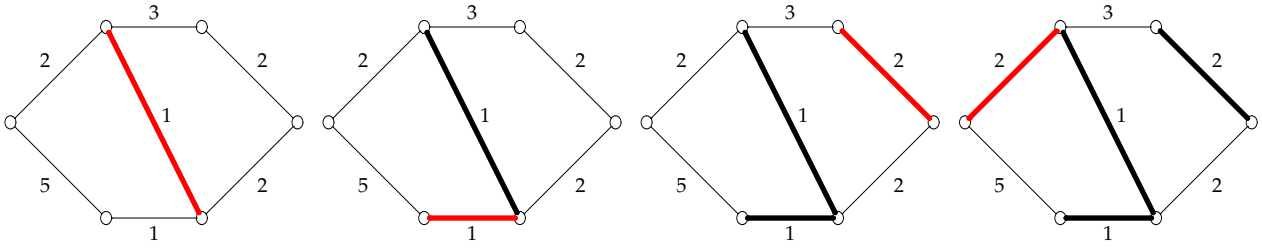
initialize: F to be empty; all edges are initially *unprocessed*

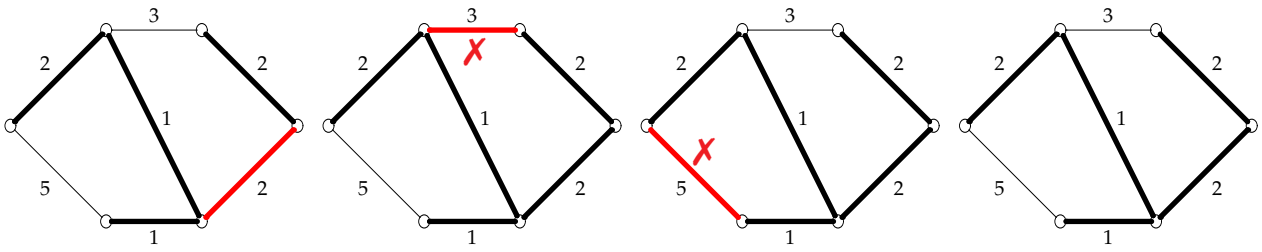
Kruskal's algorithm:

1. Find an unprocessed edge e of smallest weight w_e.
2. If $(V, F \cup \{e\})$ is a forest, then add e to F.
3. Mark e as processed and repeat until all edges have been processed.
4. Report (V, F) as a minimum-weight spanning tree.

Prim's algorithm: replace 1 by 1′

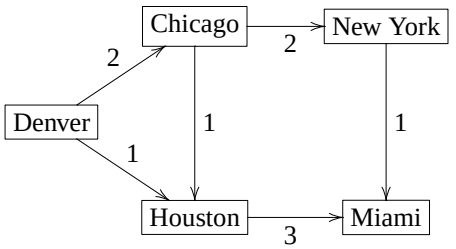
1′ Find an unprocessed edge e of smallest weight that shares an endpoint with some edge in F





10.3 Maximum Flow problem

A delivery company runs a delivery network between major US cities. Selected cities are connected by routes as shown below. On each route a number of delivery trucks is dispatched daily (indicated by labels on the corresponding edges). A customer is interested in hiring the company to deliver his products daily from Denver to Miami, and needs to know how much product can be delivered on a daily basis.



$$\begin{array}{l} \text{maximize } z \\ \\ \underbrace{\begin{array}{rrrrrrr} -x_{DC} & -x_{DH} & & & & & = -z \\ x_{DC} & & -x_{CH} & -x_{CN} & & & = 0 \\ & x_{DH} & +x_{CH} & & & -x_{HM} & = 0 \\ & & & x_{CN} & -x_{NM} & & = 0 \\ & & & & x_{NM} & +x_{HM} & = z \end{array}}_{\text{conservation of flow}} \end{array} \qquad \begin{array}{l} 0 \le x_{DC} \le 2 \\ 0 \le x_{DH} \le 1 \\ 0 \le x_{CH} \le 1 \\ 0 \le x_{CN} \le 2 \\ 0 \le x_{NM} \le 1 \\ 0 \le x_{HM} \le 3 \end{array}$$

In general, network $G = (V, E)$:

$s =$ **source** (Denver) $\qquad u_{ij} =$*capacity* of an edge ij (# trucks dispatched daily between i and j)

$t =$ **sink** (Miami) $\qquad x_{ij} =$*flow* on an edge ij (# trucks delivering the customer's products)

$$\max z$$

$$\underbrace{\sum_{\substack{j \in V \\ ji \in E}} x_{ji}}_{\text{flow into } i} - \underbrace{\sum_{\substack{j \in V \\ ij \in E}} x_{ij}}_{\text{flow out of } i} = \begin{cases} -z & i = s \\ z & i = t \\ 0 & \text{otherwise} \end{cases}$$

$$0 \le x_{ij} \le u_{ij} \qquad \text{for all } ij \in E$$

Ford-Fulkerson algorithm

Initial feasible flow $x_{ij} = 0$ for all $ij \in E$.

A sequence of nodes $v_1, v_2, \ldots, v_n$ is a *chain* if $v_i v_{i+1} \in E$ (*forward* edge) or $v_{i+1} v_i \in E$ (*backward* edge) for all $i = 1, \ldots, n-1$. If $v_1 = s$ and $v_n = t$, then we call it an (s,t)-chain. Consider an (s,t)-chain P.

The *residual capacity* of a forward edge ij on P is defined as $u_{ij} - x_{ij}$ (the remaining capacity on the edge ij). The *residual capacity* of a backward edge ij on P is defined as x_{ij} (the used capacity of the edge ij).
The *residual capacity* of P is the **minimum** taken over residual capacities of edges on P.

If the *residual capacity* of P is positive $\varepsilon > 0$, then P is an **augumenting chain**. If this happens, we can increase the flow by increasing the flow on all forward edges by ε, and decreasing the flow on all backward edges by ε. This yields a feasible flow of larger value $z + \varepsilon$. (Notice the similarity with the Transportation Problem and the ratio test in the Simplex Method – same thing in disguise.)

Optimality criterion: The flow x_{ij} is optimal if and only if there is no augmenting chain.

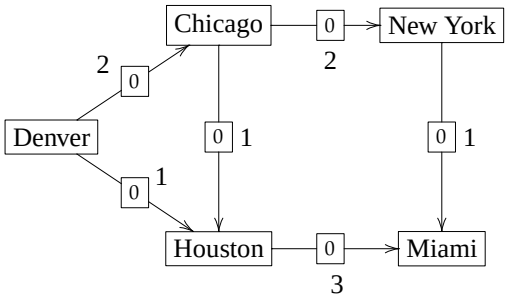
Residual network

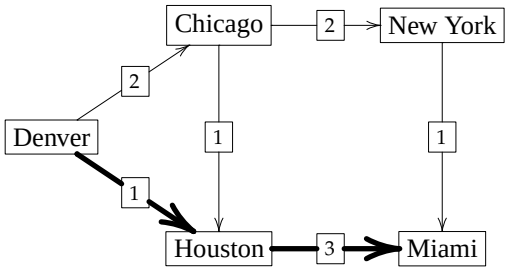
Finding an augmenting chain efficiently → residual network G_x constructed as follows:

- start by making V the nodes of G_x (no edges yet)
- then for every edge $ij \in E$,
  (a) add edge ij to G_x if $x_{ij} < u_{ij}$
  (b) add edge ji to G_x if $x_{ij} > 0$
  (if both happen then the residual network contains both the edge ij and ji)

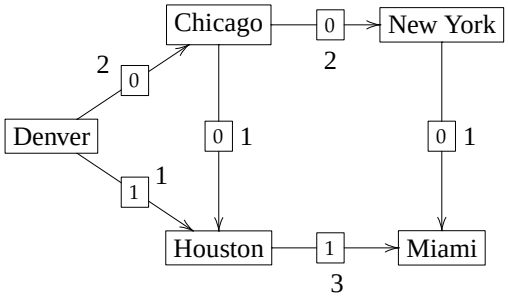
Any **path** from s to t in the residual network is an augmenting chain.

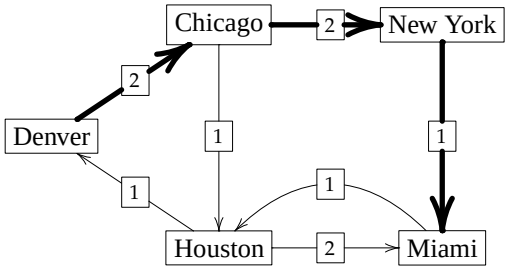
Starting feasible flow $x_{ij} = 0$ (indicated in boxes) → residual network (residual capacity shown on edges)



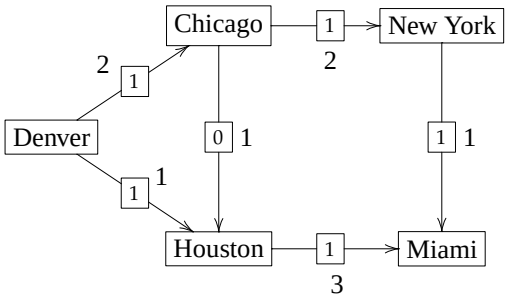


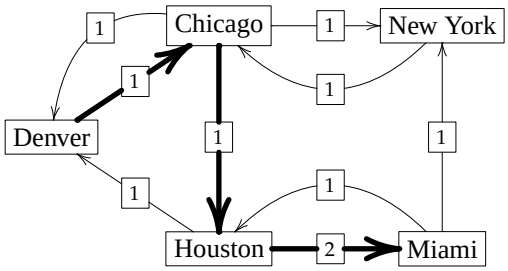
augmenting chain of residual capacity 1 → increase flow by 1



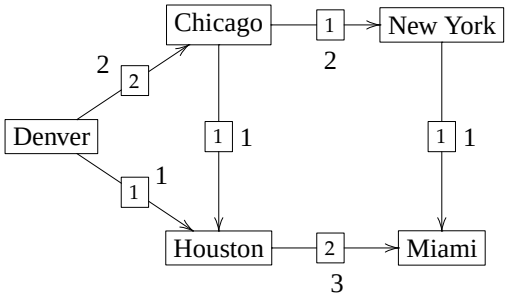


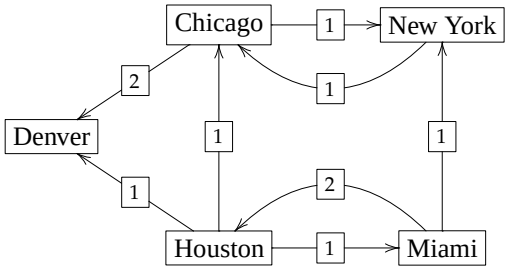
augmenting chain of residual capacity 1 → increase flow by 1





augmenting chain of residual capacity 1 → increase flow by 1





no path from Denver to Miami in the residual network → no augmenting chain → **optimal solution** found

→ maximum flow has value 3

Minimum Cut

For a subset of vertices $A \subseteq V$, the edges going between the nodes in A and the rest of the graph is called a **cut**. We write $(A, \overline{A})$ to denote this cut. The edges going out of A are called **forward edges**, the edges coming into A are **backward edges**. If A contains s but not t, then it is an (s,t)-cut.

The **capacity** of a cut $(A, \overline{A})$ is the sum of the capacities of its forward edges.

For example, let $A = \{\text{Denver,Chicago}\}$. Then $(A, \overline{A})$ is an (s,t)-cut of capacity 4. Similarly, let $A_* = \{\text{Denver, Chicago, New York}\}$. Then $(A_*, \overline{A_*})$ is an (s,t)-cut of capacity 3.

Theorem 5. *The maximum value of an (s,t)-flow is equal to the minimum capacity of an (s,t)-cut.*

This is known as the Max-Flow-Min-Cut theorem – a consequence of strong duality of linear programming.

$$\begin{array}{rrrrrrr} \text{maximize} & z & & & & & \\ -x_{DC} & -x_{DH} & & & & = & -z \\ x_{DC} & & -x_{CH} & -x_{CN} & & = & 0 \\ & x_{DH} & +x_{CH} & & -x_{HM} & = & 0 \\ & & & x_{CN} & -x_{NM} & = & 0 \\ & & & & x_{NM} & +x_{HM} = & z \end{array} \qquad \begin{array}{l} 0 \le x_{DC} \le 2 \\ 0 \le x_{DH} \le 1 \\ 0 \le x_{CH} \le 1 \\ 0 \le x_{CN} \le 2 \\ 0 \le x_{NM} \le 1 \\ 0 \le x_{HM} \le 3 \end{array}$$

Dual:

minimize $2v_{DC} + v_{DH} + v_{CH} + 2v_{CN} + v_{NM} + 3v_{HM}$

$$\begin{array}{rrrrrrr} y_D & - y_C & & & & \le & v_{DC} \\ y_D & - & y_H & & & \le & v_{DH} \\ & y_C & - y_H & & & \le & v_{CH} \\ & y_C & - & y_N & & \le & v_{CN} \\ & & & y_N & - y_M & \le & v_{NM} \\ & & & y_H & - y_M & \le & v_{HM} \\ y_D & & & & - y_M & \ge & 1 \end{array}$$

$v_{DC}, v_{DH}, v_{CH}, v_{CN}, v_{NM}, v_{HM} \ge 0$

y_D, y_C, y_H, y_N, y_M unrestricted

Optimal solution (of value 3)

$y_D = y_C = y_N = 1 \quad \rightarrow \quad A = \{D, C, N\}$ min-cut

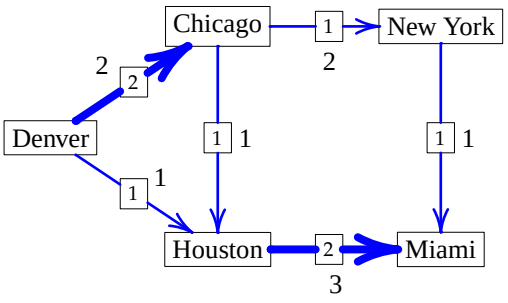
$y_H = y_M = 0$

$v_{DH} = v_{CH} = v_{NM} = 1$

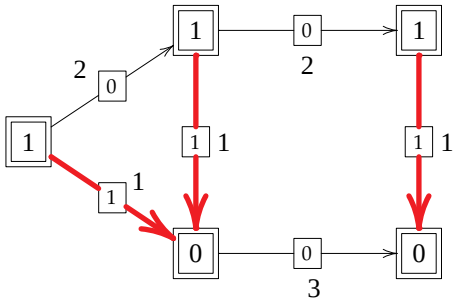
$v_{DC} = v_{CN} = v_{HM} = 0$

→ given an optimal solution, let A be the nodes whose y value is the same as that of source

→ $(A, \overline{A})$ **minimum cut**



maximum flow



minimum cut

$$\max \; z$$

$$\sum_{\substack{j \in V \\ ji \in E}} x_{ji} - \sum_{\substack{j \in V \\ ij \in E}} x_{ij} = \begin{cases} -z & i = s \\ z & i = t \\ 0 & \text{otherwise} \end{cases}$$

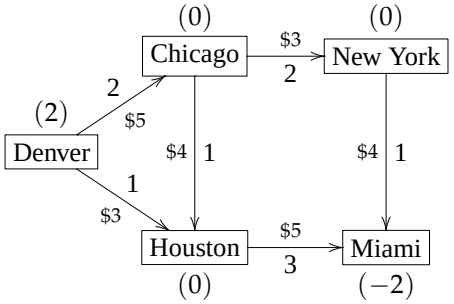
$$0 \le x_{ij} \le u_{ij} \qquad \text{for all } ij \in E$$

$$\min \sum_{ij \in E} u_{ij} v_{ij}$$

$$\begin{array}{rrrrl} y_i & - y_j & \le & v_{ij} & \text{for all } ij \in E \\ y_s & - y_t & \ge & 1 & \\ & & & v_{ij} \ge 0 & \text{for all } ij \in E \\ & & & y_i \text{ unrestricted} & \text{for all } i \in V \end{array}$$

10.4 Minimum-cost Flow problem

A delivery company runs a delivery network between major US cities. Selected cities are connected by routes as shown below. On each route a number of delivery trucks is dispatched daily (indicated by labels on the corresponding edges). Delivering along each route incurs a certain cost (indicated by the $ figure (in thousands) on each edge). A customer hired the company to deliver two trucks worth of products from Denver to Miami. What is the least cost of delivering the products?



$$
\begin{array}{l}
\text{minimize} \\
5x_{DC} + 3x_{DH} + 4x_{CH} + 3x_{CN} + 4x_{NM} + 5x_{HM}
\end{array}
\qquad
\begin{array}{ll}
0 \le x_{DC} \le 2 \\
0 \le x_{DH} \le 1 \\
0 \le x_{CH} \le 1 \\
0 \le x_{CN} \le 2 \\
0 \le x_{NM} \le 1 \\
0 \le x_{HM} \le 3
\end{array}
$$

$$
\underbrace{\begin{array}{rcl}
x_{DC}+x_{DH} & = & 2 \\
-x_{DC}+x_{CH}+x_{CN} & = & 0 \\
-x_{DH}-x_{CH}+x_{HM} & = & 0 \\
-x_{CN}+x_{NM} & = & 0 \\
-x_{NM}-x_{HM} & = & -2
\end{array}}_{\text{conservation of flow}}
$$

Minimum-cost Network Flow problem

Network $G = (V, E)$:

u_{ij} = *capacity* of an edge $(i, j) \in E$ (# trucks dispatched daily between i and j)

x_{ij} = *flow* on an edge $(i, j) \in E$ (# trucks delivering the customer's products)

c_{ij} = *cost* on an edge $(i, j) \in E$ (cost of transportation per each truck)

b_i = *net supply* of a vertex $i \in V$ (amount of products produced/consumed at node i)

$$
\min \sum_{(i,j)\in E} c_{ij}x_{ij}
$$

$$
\underbrace{\sum_{\substack{j\in V \\ ij\in E}} x_{ij}}_{\text{flow out of } i} - \underbrace{\sum_{\substack{j\in V \\ ji\in E}} x_{ji}}_{\text{flow into } i} = \underbrace{b_i}_{\text{net supply}}
$$

$$
0 \le x_{ij} \le u_{ij} \qquad \text{for all } ij \in E
$$

Necessary condition: $\sum_i b_i = 0$.

If there are no capacity constraints, the problem is called the **Transshipment problem**.

10.5 Network Simplex Algorithm

Primal

$$
\min \sum_{(i,j)\in E} c_{ij}x_{ij}
$$

$$
\sum_{\substack{j\in V \\ (i,j)\in E}} x_{ij} - \sum_{\substack{j\in V \\ (j,i)\in E}} x_{ji} = \underbrace{b_i}_{\text{net supply}} \qquad \text{for all } i \in V
$$

$$
x_{ij} \ge 0 \qquad \text{for all } (i, j) \in E
$$

Dual

$$
\max \sum_{i\in V} b_i y_i
$$

$$
y_i - y_j \le c_{ij} \quad \text{for all } (i, j) \in E
$$

$$
y_i \text{ unrestricted} \quad \text{for all } i \in V
$$

The Network Simplex method is a specialized form of the Simplex method for solving the Minimum-cost network flow problem. That is, starting from a basic feasible solution, we seek to improve the solution by increasing values

of non-basic variables. If this is not possible, we arrive at an optimal solution. If it is possible for some variable, we increase its value as much as possible until some other variable is reduced to 0 at which point we swap the two variables in our basis and repeat. To this end, we need to discuss the following issues:

1. what is basis like for the Transshipment problem?
2. how do we calculate shadow prices and reduced costs for a given basis?
3. how do we determine optimality?
4. how do we pivot – how do we find which variable leaves the basis?
5. how do we find starting feasible basis?

Basis

Note that in the Transshipment problem there are n equations, one for each vertex, but they are not independent (much like in the Transportation problem and the Maximum flow problem). However, any $n-1$ of them are independent (assuming the network is connected). So we can summarize this as follows.

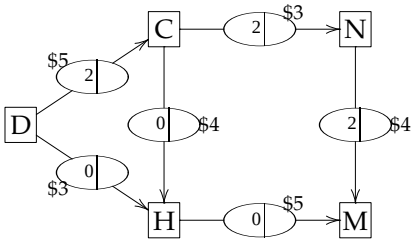
Fact 1. Every basis consist of $n-1$ basic variables.

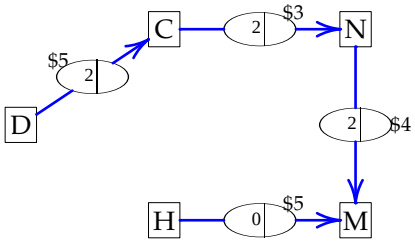
Since variables correspond to the edges of the network, this in turn implies that

Fact 2. Every basis yields a **spanning tree** of the network G.

Namely, every basis has the form $x_B = \{x_{ij} \mid (i,j) \in F\}$ where $F \subseteq E$ is set of edges of G forming a spanning tree (V,F) of G. (We ignore here the directions of edges in F.) To see this, recall that F contains exactly $n-1$ elements (since every basis has $n-1$ variables), and it cannot contain a cycle (ignoring the directions of edges), since the columns of the basis matrix B corresponding to the edges of such a cycle would not be independent (we can combine them to produce the zero vector), which is impossible, since B is invertible.

For example, consider the delivery company example without the capacities. Recall that we seek to deliver 2 units of flow (truck-loads) from Denver to Miami. An example of such a flow (**feasible** flow) is shown below. It constitutes a **basic** (feasible) solution because at most $n-1$ edges carry non-zero flow (justify this fact for yourself; from this, remember that not every feasible flow is basic). The corresponding spanning tree (**basis**) is produced by simply taking all edges with non-zero flow. If there are less than $n-1$ edges with non-zero flow (the solution is degenerate), additional edges with 0 flow might be needed to be added to form a basis – spanning tree. The choice can be arbitrary as long as there are no cycles.





Edge	Cost	Flow
DC	\$5	2
DH	\$3	0
CH	\$4	0
CN	\$3	2
NM	\$4	2
HM	\$5	0

Total cost: \$24

Shadow prices

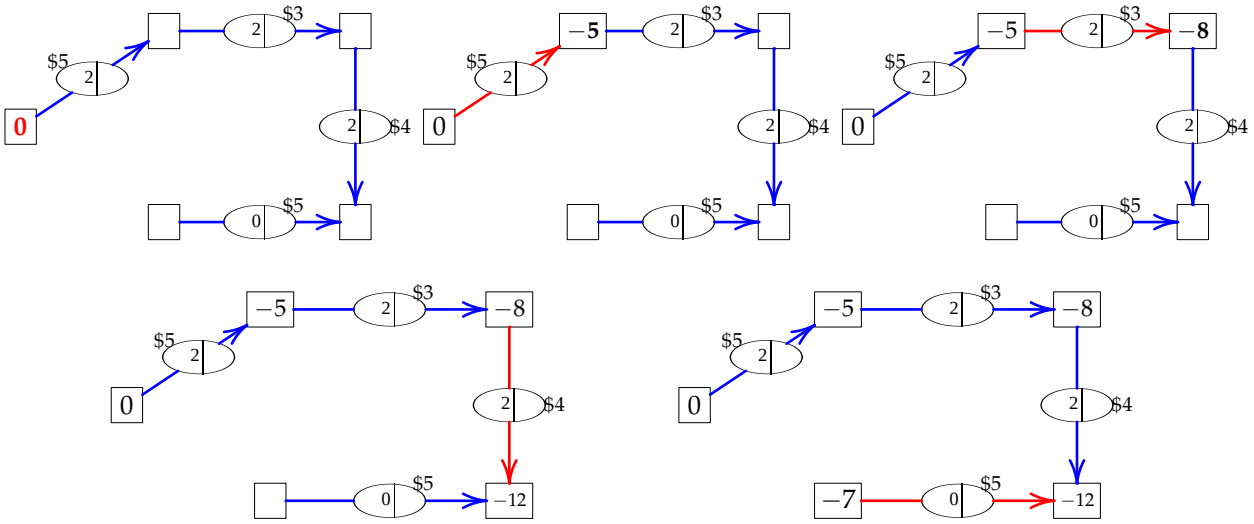
Note that shadow prices, one for each equation, correspond to nodes of G. Namely, we have a shadow price y_i for each $i \in V$ where this shadow price corresponds to the equation for conservation of flow at the node i. To find shadow prices, we use complementarity – the dual constraints corresponding to basic variables must be tight. By inspecting the dual, we see that the dual contraints are $y_i - y_j \leq c_{ij}$ for each $(i,j) \in E$. Thus for edges (i,j) forming our basis, we must satisfy this with equality, $y_i - y_j = c_{ij}$. This yields $n-1$ equations in n variables, an **underdetermined system**. To determine this system we arbitrarily set $y_1 = 0$. Then we can find all the other shadow prices as follows:

- for every basic $(i,j) \in E$:
  - if y_j has been determined but not y_i, then set $y_i = y_j + c_{ij}$;
  - if y_i has been determined but not y_j, then set $y_j = y_i - c_{ij}$.

- repeat until all shadow prices are determined.

(This should remind you of a similar process that we used in the Transportation Simplex algorithm.)

Set $y_D = \$0$. Then since the edge DC is basic, we must have $y_C = y_D - c_{DC} = y_D - \$5$. Since $y_D = \$0$, we have $y_C = -\$5$. Likewise for all other basic edges, we have $y_N = y_C - \$3$ and $y_M = y_N - \$4$ and $y_M = y_H - \$5$. Therefore $y_N = -\$8$, $y_M = -\$12$, and $y_H = -\$7$.

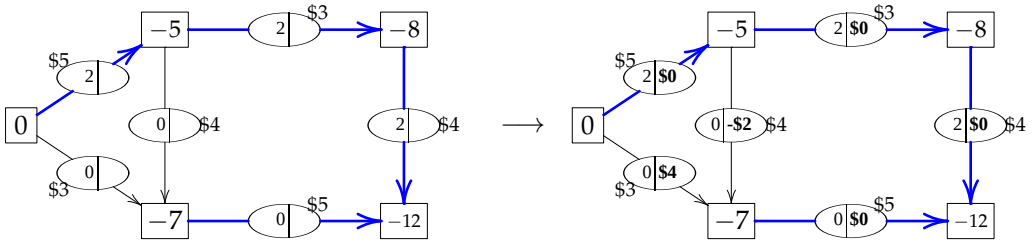


Reduced costs

Once shadow prices are found, we can determine reduced costs of all non-basic variables. We "price out" all non-basic edges. From the dual constraints, we see (since we are minimizing) that

the **reduced cost** of edge $(i, j) \in E$ is $y_i - y_j - c_{ij}$

We calculate reduced costs of every **non-basic edge** using the shadow prices we calculated above. For instance, the reduced cost of the edge DH is $y_D - y_H - c_{DH} = \$0 - (-\$7) - \$3 = \$4$. Similarly, the reduced cost of the edge CH is $y_C - y_H - c_{CH} = -\$5 - (-\$7) - \$4 = -\$2$. Note that, by definition, the reduced cost of every **basic** edge is \$0 (check this for yourself).



Optimality criterion

The solution is optimal if and only if each non-basic edge (i, j) has **non-positive** reduced cost, i.e., $y_i - y_j - c_{ij} \leq 0$

If not, then there is an edge $(i, j) \in E$ such that $x_{ij} = 0$ and $y_i - y_j > c_{ij}$. The solution can be improved by entering x_{ij} into the basis and increasing it appropriately. To do this, we need to determine the leaving variable. We do this in the following section.

In our scenario above, the edge DH has positive reduced cost \$4 while the edge CH has negative reduced cost $-\$2$. Thus the solution is not optimal and we can get a better solution by increasing the flow on the edge DH (has positive reduced cost).

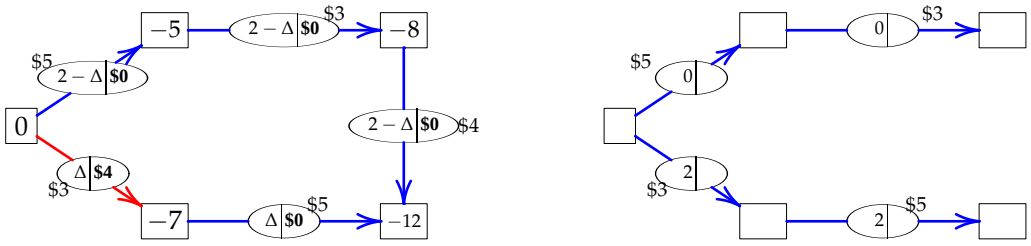
Finding a leaving variable

We change x_{ij} by value $\Delta \geq 0$. This violates conservation of flow constraints and we need to fix it. We must increase/decrease some other edges to balance the decrease/increase in x_{ij} which in turn forces some other edges to increase/decrease as well and so on. To make this work, we observe that adding (i,j) to our set of basic edges (which form a spanning tree) yields a subgraph with **exactly one** loop (cycle); this follows from the defining properties of trees (being minimally acyclic – adding any one edge creates a cycle). In particular, the cycle contains the edge (i,j). We go around this cycle and increase/decrease the flow on the edges of the cycle by Δ. Namely, if the edge is in the same direction as (i,j), we change the flow on this edge by Δ; if it is in the opposite direction, then we change it by $-\Delta$. Observe that for any value of Δ this modified flow satisfies the conservation of flow equations. It remains to find the smallest value of Δ for which the modified flow remains feasible (non-negative). Namely, we have:

(1) each edge (i',j') in the direction opposite to (i,j) on the cycle: $x_{i'j'}$ decreases to $x_{i'j'} - \Delta \geq 0$
(2) each edge (i',j') in the same direction as (i,j) on the cycle: $x_{i'j'}$ increases to $x_{i'j'} + \Delta \geq 0$

The most strict constraint of all the above determines the leaving variable. Namely, the variable giving the smallest upper bound on Δ is leaving. Then we modify the flow on the edges of the cycle by adding Δ to the flow on edges in the same direction as (i,j) and subtracting Δ to the rest.

We decided to increase the flow to Δ on the edge DH. Adding DH into the basis creates a loop D-H-M-N-C-D. We increase/decrease flow along this loop. Namely, the flow on the edge HM is increased by Δ, since this edge has the same direction (along the loop) as DH. The flow on DC, CN, NM is decreased by Δ, since the direction of these edges (with respect to the loop) is opposite to that of DH.



The largest value for Δ (guaranteeing that the flow remains non-negative on every edge) is $\Delta = 2$. With that, the flow on edges DH, HM is increased to 2, while the flow on DC, CN, NM is reduced to 0. One of these three edges is a candidate for leaving the basis. We can choose any one of DC, CN, NM (since the flow on all these three edges reduces to zero simultaneously). We choose NM. Thus we get a new basis shown above (on the right).

Economic interpretation

The value $(-y_i)$ can be interpreted as the price of the commodity (we are transporting) in node i. For instance, in the above we have $y_D = \$0$, $y_C = -\$5$ and $y_H = -\$7$. Thus the commodity costs \$0 in Denver, \$5 in Chicago, and \$7 in Houston.

It pays to transport the commodity (along an unused non-basic edge) if the **cost of transportation** is less than the **difference in price**. Namely if $y_i - y_j$ is more than c_{ij}.

For instance, the price difference between Denver and Houston is \$7 dollars and it costs \$3 to pay transportation. Thus transportation is cheaper than the price difference and consequently it pays to send along the edge DH. Comparing Chicago and Houston, we see that the price difference is \$2 and transportation costs \$4. Therefore it does not pay to send along the edge CH.

Unbounded LP

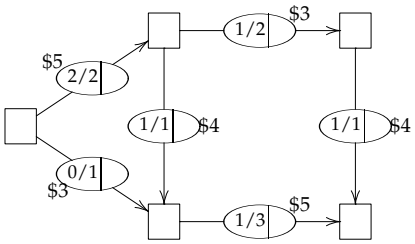
Note that it is possible that the above test finds no bound on Δ. In this case Δ can be made arbitrarily large which in turn makes the objective arbitrarily negative. In other words, the problem is unbounded. This happens precisely when $x_{ij} = 0$ and each edge on the loop is in the direction of (i, j).

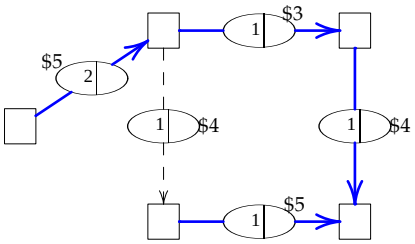
Fact 3. The problem is **unbounded** if and only if there exists a **negative cost** cycle.

10.6 Network Simplex Algorithm with capacitites

To solve the more general problem with capacities, we need to slightly modify our algorithm. In particular, the algorithm will be a variant of the Upper-Bounded Simplex method.

In addition to basic edges, we will need to keep track of **saturated edge** (edges that carry flow equal to their capacity). They will not necessarily be part of the spanning tree. We will indicate them using **dashed** lines. In addition to increasing flow on a non-basic edge (with zero flow), we will also consider **decreasing** flow on a saturated edge. This will require modifying our optimality criterion and pivotting steps.





Edge	Cost	Flow	Cap.
DC	$5	2	2
DH	$3	0	1
CH	$4	1	1
CN	$3	1	2
NM	$4	1	1
HM	$5	1	3

Total cost: $26

Optimality criterion

The solution is optimal if and only if

- each non-basic edge with **zero flow** has **non-positive** reduced costs
- each **saturated edge** has **non-negative** reduced costs.

If not, then there is either an edge $(i, j) \in E$ such that $x_{ij} = 0$ and $y_i - y_j > c_{ij}$, or there is an edge $(i, j) \in E$ such that $x_{ij} = u_{ij}$ and $y_i - y_j < c_{ij}$. The solution can be improved by entering x_{ij} into the basis and increasing/decreasing it appropriately. To do this, we need to determine the leaving variable.

Finding a leaving variable

We change x_{ij} by value Δ. Now Δ can be both positive an negative. We determine the value of Δ by considering the following inequalities.

(1) edge (i', j') in the direction opposite to (i, j) on the cycle: $0 \leq x_{i'j'} - \Delta \leq u_{i'j'}$
(2) edge (i, j): $0 \leq x_{ij} + \Delta \leq u_{ij}$
(3) edge (i', j') in the same direction as (i, j) on the cycle: $0 \leq x_{i'j'} + \Delta \leq u_{i'j'}$

The most strict constraint of all the above determines the leaving variable. Namely, if the chosen edge (i, j) carried no flow, i.e. $x_{ij} = 0$, then the variable giving the smallest upper bound on Δ is leaving (unless it is x_{ij} who gives this bound). If the chosen edge (i, j) was saturated, i.e., $x_{ij} = u_{ij}$, then the variable giving the largest lower bound leaves (unless it is x_{ij} who gives this bound). Then we modify the flow on the edges of the cycle by adding Δ to the flow on edges in the same direction as (i, j) and subtracting Δ to the rest. (Note that Δ can be negative.) If an edge becomes saturated by this, we indicate it by a dashed line in the diagrams. This is to help keep track of the current objective value.

Finding a starting feasible basis

Unlike in the Transportation problem, here we cannot use a greedy heuristic to find a starting feasible solution, since we have to, in addition, satisfy the capacity constraints (without the capacity constraints, the minimum-cost method will still work – picking edges in turn and exhausting supply on one end or saturating demand on the other).

For small problems, we can get away with trial and error. However, in general this may not be very efficient. In fact, there may not exist any feasible solution to the problem. We need a more robust method. This is provided to us in the form of the two-phase method (just like with the regular Simplex algorithm).

In Phase I, we solve an auxiliary problem whose solution will give us a starting feasible solution to our problem; then we go on from there in Phase II optimizing this solution as described earlier.

The problem in Phase I is obtained as follows. We add a new **slack** node s. For every other node $i \in V$, we

- add the edge (s, i) of capacity $-b_i$ if $b_i < 0$, or
- add the edge (i, s) of capacity b_i if $b_i \geq 0$.

The edges we added in this process we call **artificial** while the original edges are called **real** edges.

Now we assign new costs: each real edge will cost \$0, while each artificial edge will cost \$1. We call this the Phase I problem. Notice that this problem has a feasible basis, namely the basis formed by taking all artificial edges and saturating them; assigning them flow equal to their capacity.

Fact 4. The original problem has a feasible basis if and only if the Phase I problem has optimum value 0.

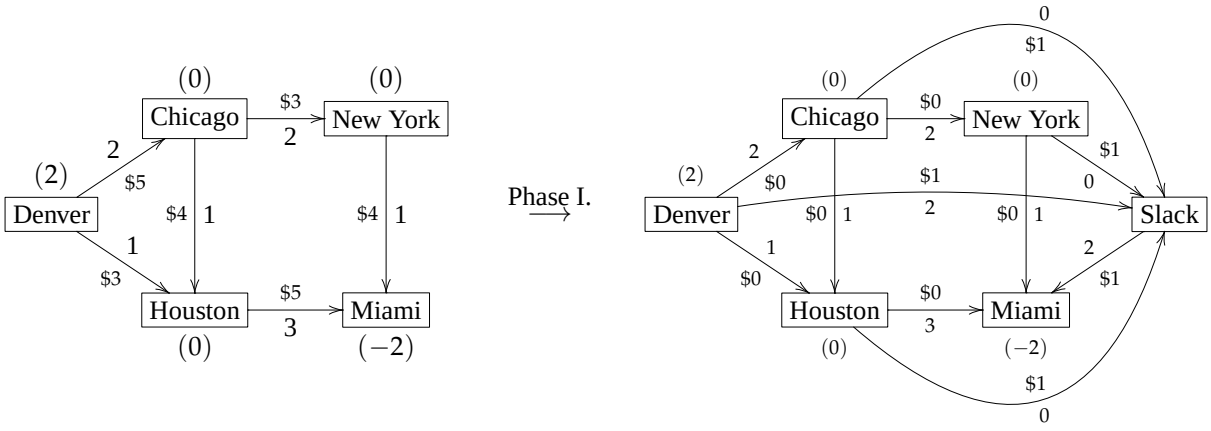
If we discover that the Phase I problem does not have optimum value 0, then this yields a set S such that

$$\sum_{i \in S} b_i > \sum_{\substack{i \in S,\, j \notin S \\ (i,j) \in E}} u_{ij}$$

Namely, the set S consists of those nodes whose shadow price is less than or equal to the shadow price of the slack node. (If we set the shadow price of the slack node to 0, then the set S simply consists of all nodes with non-positive shadow price.) This certifies that no feasible solution can exist.

10.7 Complete example

Phase I: Slack node, artificial edges to all nodes; capacities and directions of artificial edges are dictated by the net supply. If the net supply b of a node is positive (e.g. Denver), then the edge goes from this node to the slack node and capacity of this edge is b; if the net supply b is non-positive (e.g. Miami), then the edge goes from the slack node to this node and capacity is $-b$. (To remember this just note that capacity musn't be negative.) All artificial edges have cost \$1 and all other (real) edges have cost \$0.

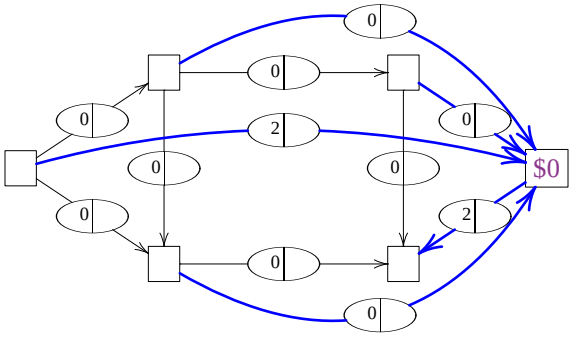


Starting basic feasible solution for Phase I: assign artificial edges flow equal to their capacity.

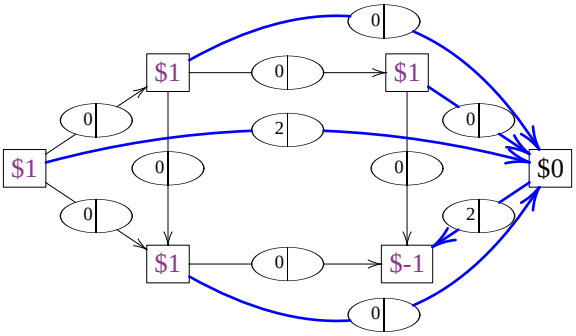
Starting basis: **all artificial edges**.

Basis shown in blue (labels on edges denote the flow and reduced costs). Everything else will not change (costs, capacities). See table below.

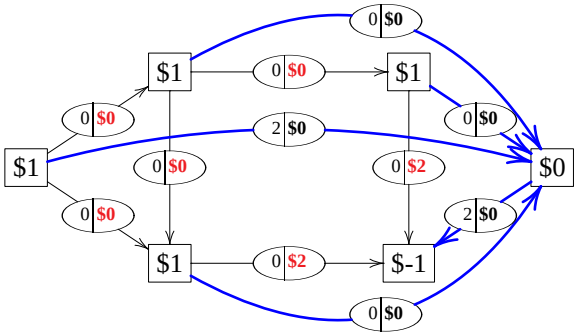
Calculate shadow prices: set $y_S = \$0$ for slack.



Knowing shadow price of Slack, we calculate shadow prices of nodes connected to Slack by basic edges; we use the fact that shadow prices satisfy $y_i - y_j = c_{ij}$ for every basic edge (i, j); if $j = S$, then the value y_j is known and so we plug it in to get y_i.



Calculate reduced costs $y_i - y_j - c_{ij}$ for all edges

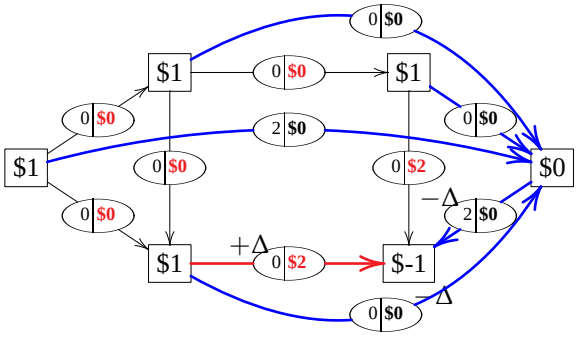


Edge	DC	DH	CH	CN	NM	HM
Cost	\$0	\$0	\$0	\$0	\$0	\$0
Flow	0	0	0	0	0	0
Capacity	2	1	1	2	1	3
Reduced cost	\$0	\$0	\$0	\$0	\$2	\$2

Edge	DS	CS	HS	NS	SM
Cost	\$1	\$1	\$1	\$1	\$1
Flow	2	0	0	0	2
Capacity	2	0	0	0	2
Reduced cost	\$0	\$0	\$0	\$0	\$0

Edge (H, M) has zero flow and positive reduced cost $\rightarrow$ adding it into basis can make the objective smaller.

Find the loop and maximum Δ.



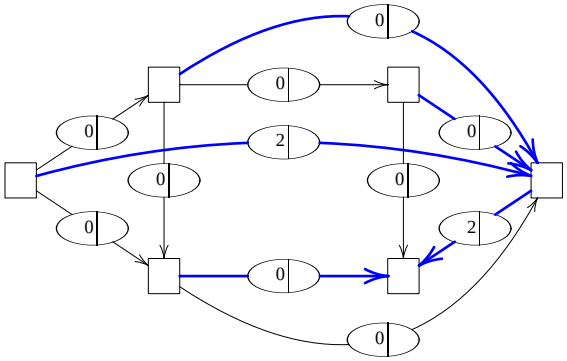
Three edges are modified:

- flow on (H, M) is increased from 0 to Δ,
- flow on (S, M) is decreased from 2 to $2 - \Delta$
- flow on (H, S) is decreased from 0 to $-\Delta$

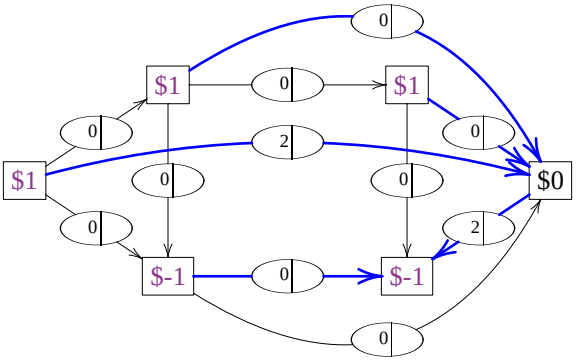
  (both edges have **opposite** direction than (H, M) on the loop)

Maximum Δ is 0 because the artificial edge (H, S) has no flow on it and thus the flow on this edge cannot decrease. The edge (H, S) leaves the basis while (H, M) enters. Flow does not change.

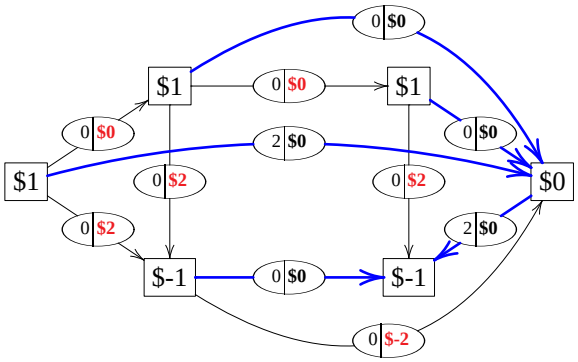
New basis:



Calculate shadow prices by setting $y_S = \$0$:

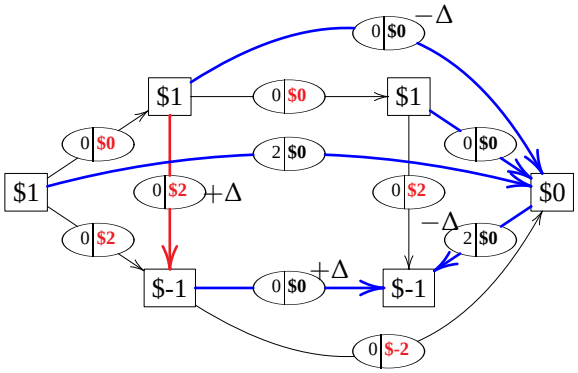


Calculate reduced costs $y_i - y_j - c_{ij}$:



Edge	DC	DH	CH	CN	NM	HM
Cost	\$0	\$0	\$0	\$0	\$0	\$0
Flow	0	0	0	0	0	0
Capacity	2	1	1	2	1	3
Reduced cost	\$0	\$2	\$2	\$0	\$2	\$0

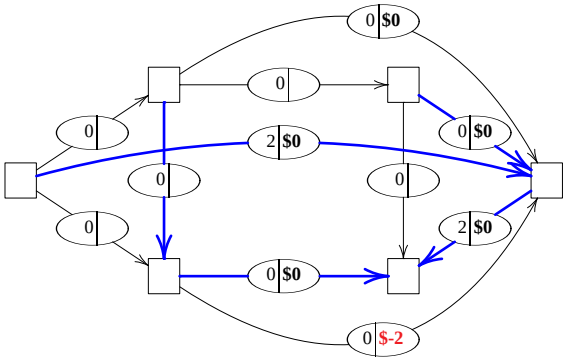
Edge	DS	CS	HS	NS	SM
Cost	\$1	\$1	\$1	\$1	\$1
Flow	2	0	0	0	2
Capacity	2	0	0	0	2
Reduced cost	\$0	\$0	\$-2	\$0	\$0



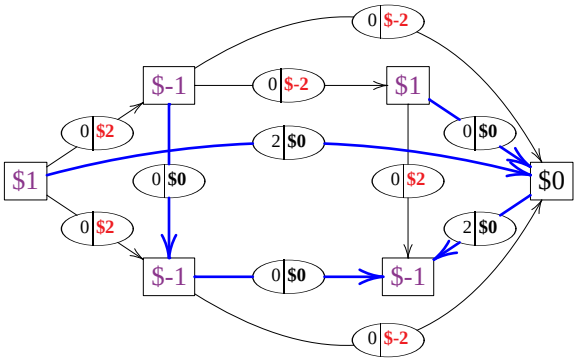
The edge (C,H) has zero flow and positive reduced cost. We increase the flow on (C,H) to Δ.

Flow on the edge (H,M) is increased to Δ (since (H,M) has same direction as (C,H) on the loop), while on (M,S) is decreased to $2-\Delta$ and on (C,S) is decreased to $-\Delta$ (both have opposite directions). Thus $\Delta = 0$ and (C,S) leaves.

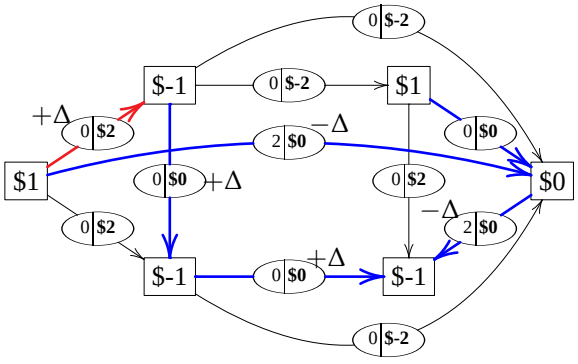
New basis:



Calculate shadow prices by setting $y_S = \$0$;
then determine reduced costs:



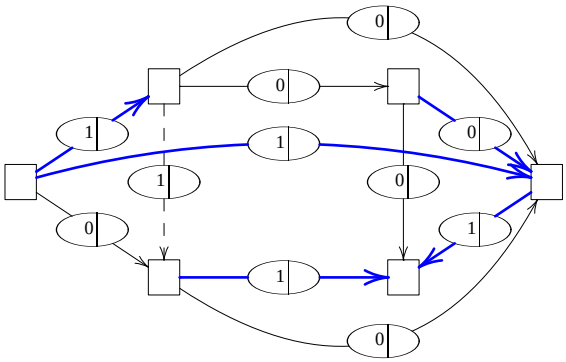
Edge (D,C) enters the basis; flow increases to Δ:



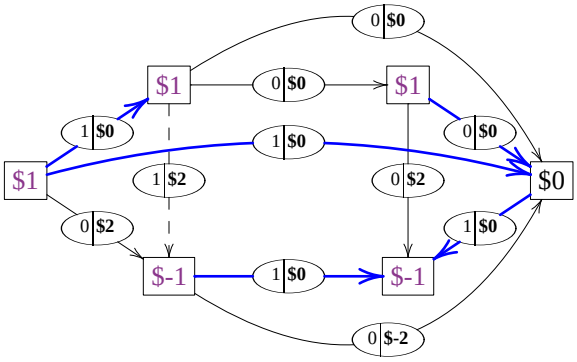
Flow increases to Δ on edges (C,H), (H,M) and decreases to $2-\Delta$ on edges (D,S) and (S,M). The capacity of (D,C) is 2, the capacity of (C,H) is 1, and of (H,M) is 3. Thus $\Delta = 1$ and (C,H) leaves the basis. This time the flow finally changes.

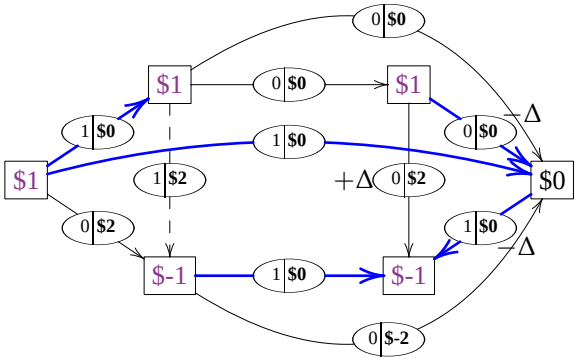
Important Note: the edge (C,H) is not in the basis. Nonetheless it carries flow. The flow on (C,H) is equal to its capacity – we say it is **saturated**.
We mark this by making the edge (C,H) dashed.

New basis:



Calculate shadow prices and reduced costs:



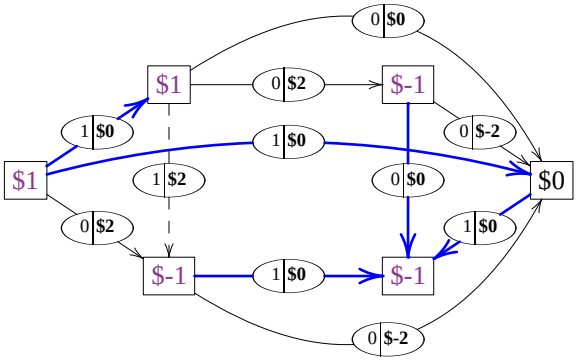


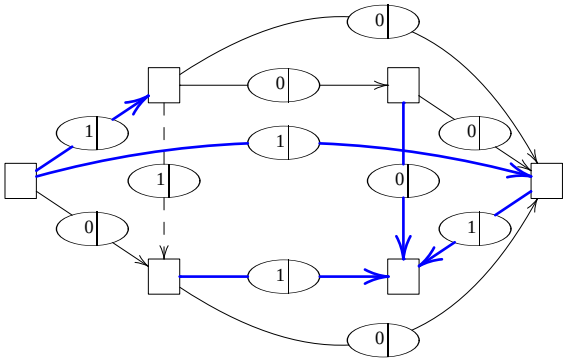
(N, M) has zero flow and positive reduced cost
$\rightarrow$ increase flow to Δ

(note that (C, H) has also positive reduced cost; however, it is saturated – saturated edges are entered if they have **negative** reduced cost)

$\Delta = 0$ and the edge (N, S) leaves the basis

Calculate shadow prices and reduced costs:

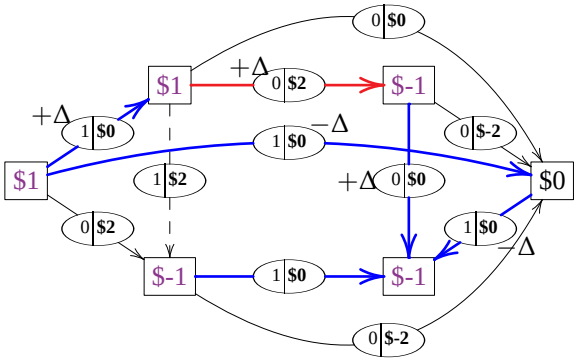


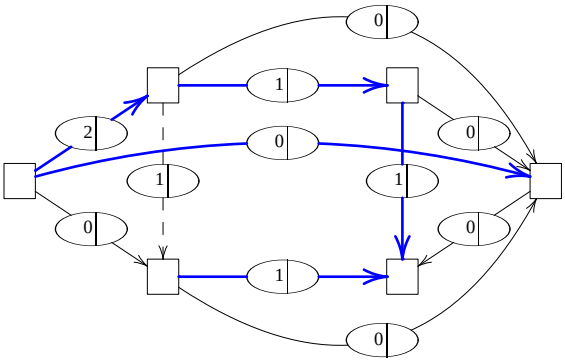


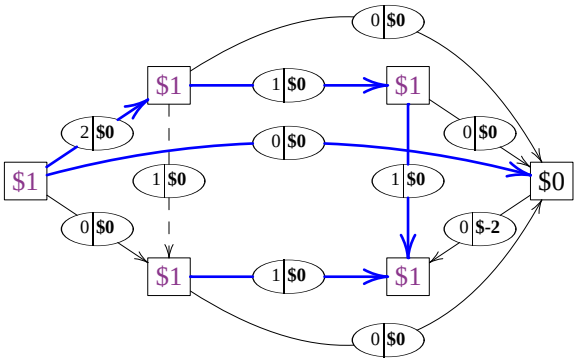
(C, N) has no flow and > 0 reduced cost $\rightarrow$ enters

Flow decreases to $1 - \Delta$ on edges (D, S) and (S, M). Flow increases to $1 + \Delta$ on edge (D, C) of capacity 2 and increases to Δ on edges (C, N) and (N, M) of capacities 2 and 1, respectively.

$\Delta = 1$ and we have a tie for the leaving variable. We arbitrarily choose (S, M) to leave. Note that flow changes this time (since $\Delta > 0$).



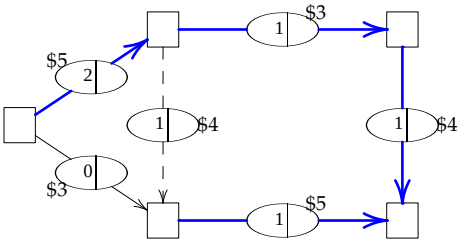




Optimal solution (of Phase I) reached:

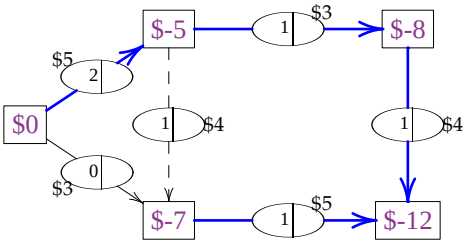
- each non-basic variable with zero flow has non-positive reduced cost.
- each saturated non-basic variable has non-negative reduced cost.

We convert this to a feasible solution to the original problem by removing the Slack node and artificial edges. Then we start Phase II. Note that from now on we use the original edge costs (unlike in Phase I where we had cost \$0 for real edges and \$1 for artificial edges).

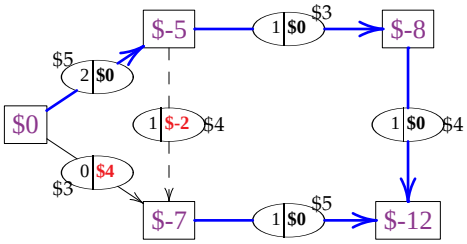


Edge	DC	DH	CH	CN	NM	HM
Cost	\$5	\$3	\$4	\$3	\$4	\$5
Flow	2	0	1	1	1	1
Capacity	2	1	1	2	1	3

Calculate shadow prices by setting $y_D = \$0$:



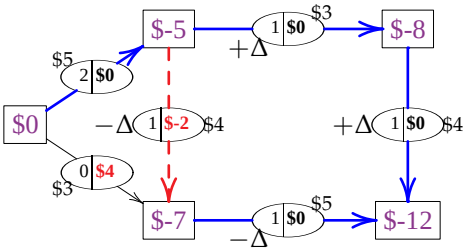
Calculate reduced costs $y_i - y_j - c_{ij}$:



Now we have two options:

- the edge (D, H) has **no flow** and **positive** reduced cost $\rightarrow$ we can increase the flow on (D, H)
- the edge (C, H) is **saturated** and has **negative** reduced cost $\rightarrow$ we can decrease the flow on (C, H)

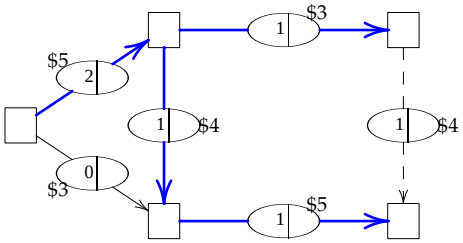
We choose (arbitrarily) do decrease the flow on (C, H) by Δ.



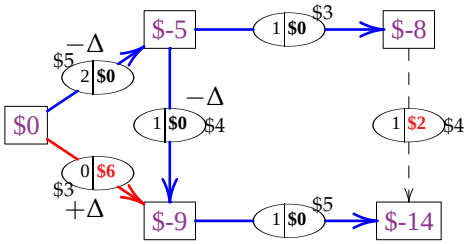
Flow on (C, H) and (H, M) changes to $1 - \Delta$.
Flow on (C, N) and (N, M) changes to $1 + \Delta$.
The capacity of (C, N) is 2, and the capacity of (N, M) is 1.

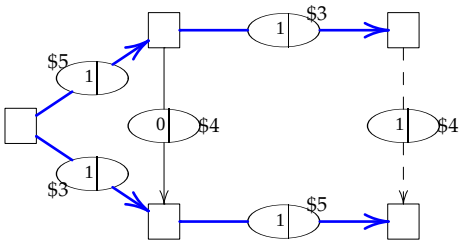
This implies that largest Δ is $\Delta = 0$, since we cannot increase the flow on (N, M) as it is already saturated

$\rightarrow$ (N, M) leaves the basis (but it stays saturated)

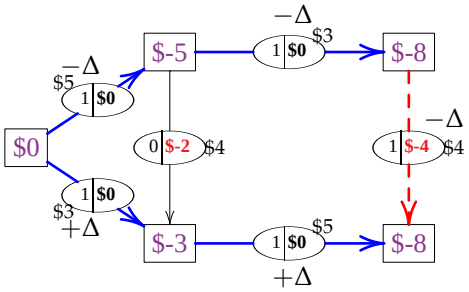


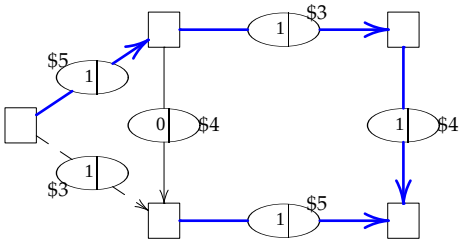
(D, H) has no flow, positive reduce cost, enters



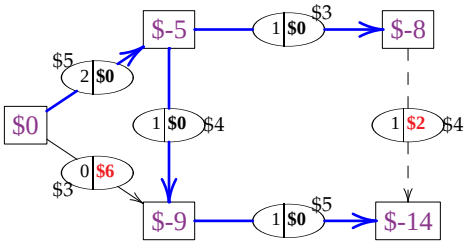


(N, M) is saturated and has negative reduced cost
$\rightarrow$ decrease flow on (N, M)





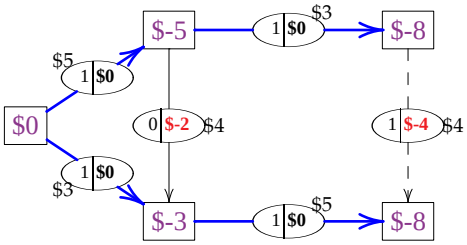
Calculate shadow prices and reduced costs:



Flow on (D, C) decreases to $2 - \Delta$. Flow on (C, H) decreases to $1 - \Delta$. Flow on (D, H) of capacity 1 increases to Δ.

$\rightarrow \Delta = 1$ and (C, H) leaves

Calculate shadow prices and reduced costs:

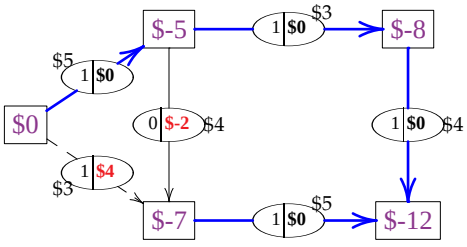


Flow on (D, C), (C, N), and (N, M) decreases to $1 - \Delta$.
Flow on (D, H) and (H, M) increases to $1 + \Delta$.
The capacity of (D, H) is 1 and the capacity of (H, M) is 3.

$\Delta = 0$ since (D, H) is already saturated

$\rightarrow (D, H)$ leaves (but stays saturated)

Calculate shadow prices and reduced costs:



Optimal solution found

- all edges with **no flow** have **non-positive** reduced cost (e.g. (C, H) has no flow and reduced cost \$-2)
- all **saturated** edges have **non-negative** reduced cost (e.g. (D, H) is saturated with reduced cost \$4)

Optimal flow consists of one unit accross all edges but (C, H). The cost is \$5+\$3+\$3+\$5+\$4=\$20.

10.8 Summary

Network $G = (V, E)$ has **nodes** V and **edges** E.

- Each edge $(i, j) \in E$ has a **capacity** u_{ij} and **cost** c_{ij}.
- Each vertex $i \in V$ provides **net supply** b_i.

For a set $S \subseteq V$, write $\overline{S}$ for $V \setminus S$ and write $E(S, \overline{S})$ for the set of edges $(i, j) \in E$ with $i \in S$ and $j \in \overline{S}$. The pair $(S, \overline{S})$ is called a **cut**. (Where applicable) there are two distinguished nodes: $s =$**source** and $t =$**sink**.

Minimum spanning tree

Primal

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$\underbrace{\sum_{(i,j)\in E(S,\overline{S})} x_{ij}}_{\text{edges from } S \text{ to } \overline{S}} > 0 \qquad \begin{array}{l}\text{for all } S \subseteq V \\ \text{where } \emptyset \neq S \neq V\end{array}$$

$$x_{ij} \geq 0 \qquad \text{for all } (i,j) \in E$$

Obstruction (to feasibility):
set $S \subseteq V$ with $\emptyset \neq S \neq V$ such that $E(S, \overline{S}) = \emptyset$

Shortest path problem

Primal

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$\underbrace{\sum_{\substack{j\in V \\ (i,j)\in E}} x_{ij}}_{\text{flow out of } i} - \underbrace{\sum_{\substack{j\in V \\ (j,i)\in E}} x_{ji}}_{\text{flow into } i} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{else} \end{cases} \qquad \text{for all } i \in V$$

$$x_{ij} \geq 0 \qquad \text{for all } (i,j) \in E$$

Dual

$$\max y_s - y_t$$

$$y_i - y_j \leq c_{ij} \qquad \text{for all } (i,j) \in E$$

$$y_i \text{ unrestricted} \qquad \text{for all } i \in V$$

Obstruction (to feasibility): set $S \subseteq V$ with $s \in S$ and $t \in \overline{S}$ such that $E(S, \overline{S}) = \emptyset$

Maximum-flow problem

Primal

$$\max z$$

$$\sum_{\substack{j\in V \\ (i,j)\in E}} x_{ij} - \sum_{\substack{j\in V \\ (j,i)\in E}} x_{ji} = \begin{cases} z & i = s \\ -z & i = t \\ 0 & \text{else} \end{cases} \qquad \text{for all } i \in V$$

$$0 \leq x_{ij} \leq u_{ij} \qquad \text{for all } (i,j) \in E$$

$$z \text{ unrestricted}$$

Dual

$$\min \sum_{(i,j)\in E} u_{ij}v_{ij}$$

$$y_i - y_j + v_{ij} \geq 0 \qquad \text{for all } (i,j) \in E$$

$$y_t - y_s = 1$$

$$v_{ij} \geq 0 \qquad \text{for all } (i,j) \in E$$

$$y_i \text{ unrestricted} \qquad \text{for all } i \in V$$

Obstruction (to feasibility): set $S \subseteq V$ with $s \in S$ and $t \in \overline{S}$ such that $z > \underbrace{\sum_{(i,j)\in E(S,\overline{S})} u_{ij}}_{\text{capacity of the cut } (S,\overline{S})}$

(no flow bigger than the capacity of a cut)

Minimum-cost (s,t)-flow problem

Primal

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$\sum_{\substack{j\in V\\(i,j)\in E}} x_{ij} - \sum_{\substack{j\in V\\(j,i)\in E}} x_{ji} = \begin{cases} f & i=s \\ -f & i=t \\ 0 & \text{else} \end{cases} \quad \text{for all } i \in V$$

$$0 \le x_{ij} \le u_{ij} \quad \text{for all } (i,j) \in E$$

Dual

$$\max fy_s - fy_t - \sum_{(i,j)\in E} u_{ij}v_{ij}$$

$$y_i - y_j - v_{ij} \le c_{ij} \quad \text{for all } (i,j) \in E$$

$$v_{ij} \ge 0 \quad \text{for all } (i,j) \in E$$

$$y_i \text{ unrestricted} \quad \text{for all } i \in V$$

Obstruction (to feasibility): set $S \subseteq V$ with $s \in S$ and $t \in \overline{S}$ such that $f > \sum_{(i,j)\in E(S,\overline{S})} u_{ij}$

Transshipment problem

Primal

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$\sum_{\substack{j\in V\\(i,j)\in E}} x_{ij} - \sum_{\substack{j\in V\\(j,i)\in E}} x_{ji} = \underbrace{b_i}_{\text{net supply}} \quad \text{for all } i \in V$$

$$x_{ij} \ge 0 \quad \text{for all } (i,j) \in E$$

Dual

$$\max \sum_{i\in V} b_i y_i$$

$$y_i - y_j \le c_{ij} \quad \text{for all } (i,j) \in E$$

$$y_i \text{ unrestricted} \quad \text{for all } i \in V$$

Obstruction (to feasibility): set $S \subseteq V$ such that $\sum_{i\in S} b_i > 0$ and $E(S,\overline{S}) = \emptyset$

Minimum-cost network flow problem

Primal

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$\sum_{\substack{j\in V\\(i,j)\in E}} x_{ij} - \sum_{\substack{j\in V\\(j,i)\in E}} x_{ji} = b_i \quad \text{for all } i \in V$$

$$0 \le x_{ij} \le u_{ij} \quad \text{for all } (i,j) \in E$$

Dual

$$\max \sum_{i\in V} b_i y_i - \sum_{(i,j)\in E} u_{ij}v_{ij}$$

$$y_i - y_j - v_{ij} \le c_{ij} \quad \text{for all } (i,j) \in E$$

$$v_{ij} \ge 0 \quad \text{for all } (i,j) \in E$$

$$y_i \text{ unrestricted} \quad \text{for all } i \in V$$

Obstruction (to feasibility): set $S \subseteq V$ such that $\sum_{i\in S} b_i > \sum_{(i,j)\in E(S,\overline{S})} u_{ij}$

11

Game Theory

During the 8PM to 9PM time slot, two TV networks compete for an audience of 100 million viewers. The networks announce their schedule ahead of time and do not know of each other's decision until the show time. Based on that a certain number of people will tune to Network 1 while the rest will watch Network 2. The market research revealed the following expected number of viewers of Network 1.

	Network 2		
Network 1	Western	Soap Opera	Comedy
Western	35	15	60
Soap Opera	45	58	50
Comedy	38	14	70

For instance, if Network 1 shows a Western while Network 2 shows a Comedy, then

60 million viewers will watch Network 1, and

$100 - 60 = 40$ million watch Network 2.

Question: What strategy should the two networks use to maximize their viewership?

Terminology:

- Network 1 is a **row player**.
- Network 2 is a **column player**.
- The above matrix is called a **payoff matrix**.
- This is a **constant-sum** game (the outcome for both players always sums up to a constant 100 million).

How to solve this game? Let us look at the structure of outcomes. For instance, if Network 1 chooses to show Western, then it can get as many as 60 million viewers (if Network 2 chooses to show a Comedy) but also as little as 15 million (if Network 2 shows a Soap Opera). Thus this choice cannot guarantee more than 15 milion viewers for Network 1 (in the worst case). If the network instead chooses to show a Comedy, the situation is even worse, since then we can guarantee only 14 million viewers (the minimum in the 3rd row). The best therefore for Network 1 is to choose to show a Soap Opera in which case 45 million or more viewers will tune to Network 1 regardless of what Network 2 does. Note that in this strategy Network 1 (being the row player) simply calculates the **row minimum** of each row and then chooes the row with **largest** row minimum.

By the same argument, Network 2 (being the column player) can maximize its vieweship by calculating each **column maximum** and choosing column with the smallest column maximum.

It is easy to see that the two outcomes will satisfy the following inequality

$$\max_{\text{all rows}} \ (\text{row minimum}) \ \leq \ \min_{\text{all columns}} \ (\text{column maximum})$$

In this example, Network 1 chooses Soap Opera and Network 2 chooses Western whereby 45 million viewers will watch Network 1 and 55 million will watch Network 2. Note that this choice is simultaneously best for both Network

1 and Network 2 (we have equality $\max(\text{row minimum}) = \min(\text{col maximum})$). This is called a **saddle point** and the common value of both sides of the equation is called the **value** of the game.

- An **equilibrium point** of the game: choice of strategies for both players such that neither player can improve their outcome by changing his strategy.
- A **saddle point** of a game is an example of an equilibrium.

11.1 Pure and Mixed strategies

Note the in the above example each player's strategy was deterministic; they each examined possible outcomes and made a specific single choice to follow. This is called a **pure strategy**.

Unfortunately, there are games with no saddle points and following a pure strategy may not always give the players the best outcome. (Consider for instance Poker – see next section.) We need to generalize our definition of strategy. Instead of choosing one fixed move, the player considers all moves and chooses randomly according to a distribution. The outcome of the game will depend on the chosen distribution.

- a **mixed (randomized) strategy** $(x_1, x_2, \ldots, x_n)$ is a probability distribution over n possible moves of the player (i.e. satisfies $x_1 + x_2 + \ldots + x_n = 1$)
- **optimal strategy** is a strategy maximizing the **expected** gain for the player

Note that since we are now using chance we can sometimes gain more, sometimes less; thus we will settle for maximizing the average outcome (if, say, we play the game repeatedly for many rounds).

Betting Game

Player 1 draws a card from a card deck (and hides it from the other player). Then he decides to either **pass** in which case he discards the card and pays \$1 to Player 2, or he will **bet** in which case it is 2nd player's turn. Player 2 can either **fold** in which case she pays \$1 to Player 1, or she will **call** and the card is revealed.

If the revealed card is **high** (10, Jack, Queen, King, Ace), then Player 2 pays \$2 to Player 1. Otherwise, the card is **low** (2 through 9) and Player 1 pays \$2 to Player 2.

Let us analyze the possible strategies for Player 1. Given the card, the card can be either high or low. Based on that the player can either pass or bet. So there are 4 possible strategies: pass on both high and low (**PP**), pass on high and bet on low (**PB**), bet on high and pass on low (**BP**), and bet on both high and low (**BB**). Player 2 can either **Call** or **Fold**. The possible expected outcomes of the game are then as follows.

Player 1	Player 2		
	Call	**Fold**	Row Minimum
PP	-1	-1	-1
PB	$-1\frac{8}{13}$	$\frac{3}{13}$	$-1\frac{8}{13}$
BP	$\frac{2}{13}$	$-\frac{3}{13}$	$-\frac{3}{13}$
BB	$-\frac{6}{13}$	1	$-\frac{6}{13}$
Column Maximum	$\frac{2}{13}$	1	

Thus, for instance, suppose that Player 1 plays strategy **BP** (bet on high, pass on low) and Player 2 **calls**. Then either the card is high, Player 1 bets and wins \$2, or the card is low and Player 1 folds and loses \$1. The chance of getting a high card is $5/13$ and thus getting a low card has probability of $8/13$. So on average Player 1 gains

$$\frac{5}{13}\$2 + \frac{8}{13}(-\$1) = \frac{2}{13} \approx \$0.15$$

On the other hand, if Player 1 plays **BP** but Player 2 **folds**, then the expected gain of Player 1 is only $-\frac{3}{13} \approx -\$0.23$. So clearly, the intuitive strategy **BP** (bet on high, pass on low) is not necessarily best against a worst-case opponent.

Note that this is a **zero-sum** game since either Player 1 pays Player 2 or vice-versa (the sum of the players' gains is zero). Looking at the table above, we see that largest row minimum is $-\frac{3}{13}$ while smalest column maximum is $\frac{2}{13}$. So this game does not have a saddle point (unlike the first game we discussed).

Notice that some strategies are better then others in all possible cases. For instance, playing **BP** instead of **PP** always gives better outcome for Player 1. We say that the strategy **PP** is **dominated** by another strategy (in this case **BP**). Clearly, if a strategy is dominated, the player can always do better by playing another strategy. Thus we can safely remove any dominated strategy from consideration without changing the problem (its optimal solution). In our case, this eliminates the strategy **PP** as well as the strategy **PB** which is dominated by the strategy **BB**. This leaves us with only two strategies and the pay-off matrix shown on the right.

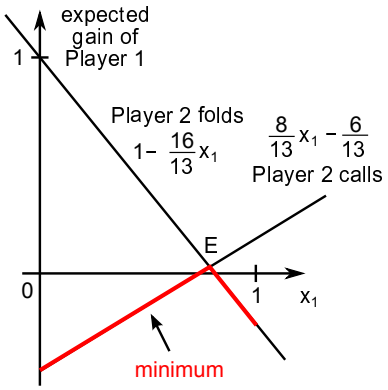
	Player 2	
Player 1	**Call**	**Fold**
BP	$\frac{2}{13}$	$-\frac{3}{13}$
BB	$-\frac{6}{13}$	1

Now we are ready to determine the **best mixed strategy** for Player 1. The player chooses to play **BP** with probability x_1 or plays **BB** with probability x_2, where $x_1 + x_2 = 1$. We denote this **mixed strategy** as (x_1, x_2). An **expected outcome** of this strategy is $\frac{2}{13}x_1 - \frac{6}{13}x_2$ if Player 2 **calls** and is $-\frac{3}{13}x_1 + x_2$ if Player 2 **folds**. So the worst-case outcome is simply the minimum of the two

$$\min_{(x_1,x_2)} \{\tfrac{2}{13}x_1 - \tfrac{6}{13}x_2, -\tfrac{3}{13}x_1 + x_2\}$$

Since $x_1 + x_2 = 1$, we can simplify this to

$$\text{outcome} = \min_{x_1}\{\tfrac{8}{13}x_1 - \tfrac{6}{13}, 1 - \tfrac{16}{13}x_1\}$$



We can plot the possible outcomes (based on the choices of x_1) as shown on the right. From this we determine that the best mixed strategy for Player 1 is the point E corresponding to strategy (x_1, x_2) where $x_1 = 19/24$ and $x_2 = 5/24$. This guarantees the player expected gain of $1/39 \approx \$0.025$.

Answer: The best strategy for Player 1 is to randomly choose between betting on high, passing on low with probability $79\frac{1}{6}\%$ or always bet with probability $20\frac{5}{6}\%$. This gives him expected gain of ¢2.5.

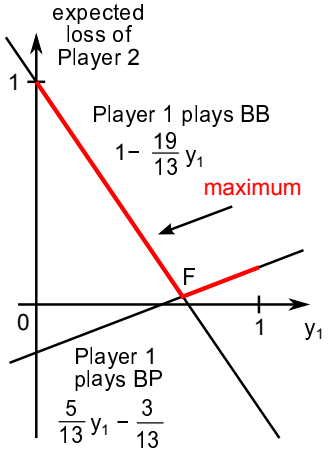
Similarly, we can determine the best mixed strategy for Player 2. Let (y_1, y_2) be the probabilities of Player 2 calling (y_1) or folding (y_2) where $y_1 + y_2 = 1$. If Player 1 plays **BP**, then the expected outcome (loss) for Player 2 is $\frac{2}{13}y_1 - \frac{3}{13}y_2$. If Player 1 play **BB**, then the outcome is $-\frac{6}{13}y_1 + y_2$. Thus the worst-case loss for Player 2 is the maximum of the two.

$$\max_{(y_1,y_2)} \{\tfrac{2}{13}y_1 - \tfrac{3}{13}y_2, -\tfrac{6}{13}y_1 + y_2\}$$

Using the fact that $y_1 + y_2 = 1$, this simplifies to

$$\max_{y_1}\{\tfrac{5}{13}y_1 - \tfrac{3}{13}, 1 - \tfrac{19}{13}y_1\}$$

We can plot the possible outcomes as shown on the right. The best strategy for Player 2 is the point F corresponding to strategy (y_1, y_2) where $y_1 = \frac{2}{3}$ and $y_2 = \frac{1}{3}$ with expected loss of $\frac{1}{39} \approx \$0.025$. Thus the best strategy for Player 2 is to call with $66\frac{2}{3}\%$ probability and to fold with $33\frac{1}{3}\%$ probability.



Linear programing

We can now see what are the best strategies for the two players. Player 1 tries to choose probabilities x_1, x_2 so as to maximize the $\min\{\frac{2}{13}x_1 - \frac{6}{13}x_2, -\frac{3}{13}x_1 + x_2\}$ where probabilities x_1, x_2 sum up to 1. Similarly, Player 2 chooses his probabilities y_1, y_2 so as to minimize $\max\{\frac{2}{13}y_1 - \frac{3}{13}y_2, -\frac{6}{13}y_1 + y_2\}$ where $y_1 + y_2 = 1$. Using the tricks we learnt, we see that both these problem can be transformed into linear programs as follows.

$$\begin{array}{rrcrcl} \max z & & & & & \\ \text{subject to} & \frac{2}{13}x_1 & - & \frac{6}{13}x_2 & \geq & z \\ & -\frac{3}{13}x_1 & + & x_2 & \geq & z \\ & x_1 & + & x_2 & = & 1 \\ & & & x_1, x_2 & \geq & 0 \end{array} \qquad \begin{array}{rrcrcl} \min w & & & & & \\ \text{subject to} & \frac{2}{13}y_1 & - & \frac{3}{13}y_2 & \leq & w \\ & -\frac{6}{13}y_1 & + & y_2 & \leq & w \\ & y_1 & + & y_2 & = & 1 \\ & & & y_1, y_2 & \geq & 0 \end{array}$$

which we can rewrite as follows:

$$\begin{array}{rrcrcrcl} \max & & & & & z & & \\ \text{subject to} & -\frac{2}{13}x_1 & + & \frac{6}{13}x_2 & + & z & \leq & 0 \\ & \frac{3}{13}x_1 & - & x_2 & + & z & \leq & 0 \\ & x_1 & + & x_2 & & & = & 1 \\ & & & & & x_1, x_2 & \geq & 0 \\ & & & & & z & \text{unrestricted} & \end{array} \qquad \begin{array}{rrcrcrcl} \min & & & & & w & & \\ \text{subject to} & -\frac{2}{13}y_1 & + & \frac{3}{13}y_2 & + & w & \geq & 0 \\ & \frac{6}{13}y_1 & - & y_2 & + & w & \geq & 0 \\ & y_1 & + & y_2 & & & = & 1 \\ & & & & & y_1, y_2 & \geq & 0 \\ & & & & & w & \text{unrestricted} & \end{array}$$

Notice that the two programs are **duals** of each other. This is not a coincidence. In fact, it is always the case in zero-sum (constant-sum) games. This tells us that the optimal solutions to the two programs have the **same value** by Strong Duality, which defines the **value** of the game. Moreover, the optimal strategies for the two players satisfy Complementary Slackness (verify this for the solutions we found above). In other words, the solutions to the two problems form an **equilibrium** point (neither play can do better by changing his/hers strategy). In the literature, this is often called **Nash equilibrium**.

11.2 Nonconstant-sum Games

In the real world, more often we find situations where the gains/losses of the players are not necessarily constant. This happens for instance in cases where cooperating players can gain more together than by competing alone.

A prototypical example of this is the famous **Prisoners' dilemma**. In this problem, we have two criminals who commited a crime but there is not enough evidence against both of them. The district attorney gives each a chance to confess. If both confess, then they both go to jail for 5 years. If only one of them confesses, he is let free and the other criminal gets 20 years in jail. If neither confesses, they are each sentenced for 1 year for a misdemeanor. The two criminals are not allowed to communicate. **What is the best strategy** for each of them?

	Criminal 2	
Criminal 1	**Confess**	**Don't confess**
Confess	$(-5,-5)$	$(0,-20)$
Don't confess	$(-20,0)$	$(-1,-1)$

The pay-off matrix now has different pay-offs for the two players as shown on the right. Clearly the pay-offs do not sum up to the same value each time. If they both confess they each go to jail for 5 years (sum is 10), while if they both do not confess, they each go to jail only for 1 year (sum is 2).

Note that both confessing is an **equilibrium** point, since if only one of them changes his mind (does not confess), he goes to jail for 20 year (which is more than 5 he gets by confessing). However, if they both change their mind (do not confess), they both improve their situation (each only gets 1 year in jail). This does not change our conclusion that confessing is an equilibrium. To be an equilibrium we only check that each player by himself cannot get a better outcome if he changes his mind (and others play the same way.

On the other hand, both not confessing, is not an equilibrium, since either player can change his mind and confess and thus not go to jail (while the other player gets 20 years – again, we only consider a single player changing his mind).

To solve this game, we see that the strategy **Don't confess** is **dominated** by strategy **Confess** in both players (the player can always do better by confessing when playing against a worst-case opponent). By eliminating these strategies, we are left with the unique equilibrium where both players confess.

Recall that this is a **pure** (deterministic) strategy. In other cases, eliminating dominated strategies will leave more than one choice and we formulate the maximization problem just like we did before. Solving it for both players gives us their optimal **mixed strategy**.

12 Integer programming

Integer Linear Program

$$\begin{aligned} \max\ & \mathbf{cx} \\ & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \\ & \mathbf{x} \text{ integer} \end{aligned}$$

- **pure** IP = all variables integer
- **mixed** IP = some variables integer
- **LP relaxation** of an ILP = dropping the integer restriction

Example:

(PURE) INTEGER PROGRAM	MIXED INTEGER PROGRAM	LP RELAXATION
$\max 4x_1 + 2x_2 + 3x_3$	$\max 4x_1 + 2x_2 + 3x_3$	$\max 4x_1 + 2x_2 + 3x_3$
$5x_1 + 3x_2 + 4x_3 \leq 7$	$5x_1 + 3x_2 + 4x_3 \leq 7$	$5x_1 + 3x_2 + 4x_3 \leq 7$
$x_1, x_2, x_3 \in \{0,1\}$	$0 \leq x_1, x_2 \leq 1$	$0 \leq x_1, x_2, x_3 \leq 1$
	$x_3 \in \{0,1\}$	
optimal solution:	optimal solution:	optimal solution:
$x_1 = 0, x_2 = x_3 = 1, z = 5$	$x_1 = 0.6, x_2 = 0, x_3 = 1, z = 5.4$	$x_1 = 1, x_2 = 0, x_3 = 0.5, z = 5.5$

12.1 Problem Formulation

Review LP formulations

Standard LP problems:
- diet problem
- product mix
- blending
- inventory
- scheduling
- budgeting

Tricks:

1. maximum/minimum:

$$\max\{f,g\} \leq h \iff \text{2 constraints } \begin{aligned} f &\leq h \\ g &\leq h \end{aligned}$$

$$\min\{f,g\} \geq h \iff \text{2 constraints } \begin{aligned} f &\geq h \\ g &\geq h \end{aligned}$$

2. absolute value: $|f-g| \leq h \iff$ 2 constraints $\begin{aligned} f - g &\leq h \\ g - f &\leq h \end{aligned}$

   (to see this, note that $|f-g| \leq h$ means that g lies between $f-h$ and $f+h$; that is, $f-h \leq g \leq f+h$)

3. positive/negative values $f = x^+ - x^-$ where x^+, x^- are new non-negative variables

   $\Rightarrow$ now x^+ is the positive part of f (or zero) and x^- the negative part of f (or zero)

   Important: must make sure that x^+ and x^- are never both basic in the optimum, else it doesn't work

   alternatively: 2 constraints $f \leq x^+$ and $-f \leq x^-$ (with similar caveats)

Standard problems

Knapsack - resource allocation, portfolio selection

– 4 possible investments
– $14,000 cash available

	investment	yield
1	$5,000	$16,000
2	$7,000	$22,000
3	$4,000	$12,000
4	$3,000	$8,000

$$\max 16x_1 + 22x_2 + 12x_3 + 8x_4$$
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$
$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

optimal solution: $z = \$42,000$
$x_1 = 0$, $x_2 = x_3 = x_4 = 1$

(burglar Bill wants to steal items of as much value as possible, but has has limited size knapsack)

LP relaxation of the Knapsack problem is called **Fractional Knapsack**
– optimal solution: pick items with highest value per unit

$$\max 16x_1 + 22x_2 + 12x_3 + 8x_4$$
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$
$$0 \leq x_1, x_2, x_3, x_4 \leq 1$$

	investment	yield/$1
1	$5,000	$3.20
2	$7,000	$3.14
3	$4,000	$3.00
4	$3,000	$2.67

optimal solution: $z = \$44,000$
$x_1 = x_2 = 1$, $x_3 = 0.5$, $x_4 = 0$

More constraints:

'at most two investments' $\iff$ $x_1 + x_2 + x_3 + x_4 \leq 2$

'if invested in #2, then must invest in #3' $\iff$ $x_3 \geq x_2$

'if invested in #1, then cannot invest in #4' $\iff$ $x_1 + x_4 \leq 1$

Fixed-charge problem – facility location

- company needs to serve 3 regions with weekly demands 80, 70, and 40 units, respectively.
- can open a facility in any of 4 cities (New York, Los Angeles, Chicago, Atlanta)
- each facility can ship 100 units (per week)
- opening a facility caries operating costs (per week)

		Transportation costs (per unit)		
Facility	Cost	Region 1	Region 2	Region 3
New York	$400	$20	$40	$50
Los Angeles	$500	$48	$15	$26
Chicago	$300	$26	$36	$18
Atlanta	$150	$24	$50	$35

x_{ij} = amount transported from facility i to region j

$y_i \in \{0,1\}$ = indicates if facility is opened (1) in city i or not (0)

$$\min \quad 400y_1 + 20x_{11} + 40x_{12} + 50x_{13} + 500y_2 + 48x_{21} + 15x_{22} + 26x_{23}$$
$$+300y_3 + 26x_{31} + 36x_{32} + 18x_{33} + 150y_4 + 24x_{41} + 50x_{42} + 35x_{43}$$

$$x_{11} + x_{21} + x_{31} + x_{41} \geq 80$$
$$x_{12} + x_{22} + x_{32} + x_{42} \geq 70$$
$$x_{13} + x_{23} + x_{33} + x_{43} \geq 40$$

$$x_{11} + x_{12} + x_{13} \leq 100y_1$$
$$x_{21} + x_{22} + x_{23} \leq 100y_2$$
$$x_{31} + x_{32} + x_{33} \leq 100y_3$$
$$x_{41} + x_{42} + x_{43} \leq 100y_4$$

$$y_1, y_2, y_3, y_4 \in \{0,1\}$$
$$x_{ij} \geq 0 \text{ for } i \in \{1,2,3,4\}, \quad j \in \{1,2,3\}$$

Set-cover problem

- open facilities to serve 6 cities
- a facility in one city can serve neighbouring cities within 15 miles

Distance	City 2	City 3	City 4	City 5	City 6
City 1	10	20	30	30	20
	City 2	25	35	20	10
		City 3	15	30	20
			City 4	15	25
				City 5	14

City	1	2	3	4	5	6
Cost	$500	$340	$450	$250	$200	$300

$x_i \in \{0,1\}$ = indicates if a facility is opened in city i

City	1	2	3	4	5	6
Neighbouring cities	1,2	1,2,6	3,4	3,4,5	4,5,6	2,5,6

$$\min z = 500x_1 + 340x_2 + 450x_3 + 250x_4 + 200x_5 + 300x_6$$

$$\begin{aligned} x_1 + x_2 &\geq 1 \\ x_1 + x_2 + x_6 &\geq 1 \\ x_3 + x_4 &\geq 1 \\ x_3 + x_4 + x_5 &\geq 1 \\ x_4 + x_5 + x_6 &\geq 1 \\ x_2 + x_5 + x_6 &\geq 1 \end{aligned}$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

Non-linear objectives – price discounts (piece-wise linear functions)

- first 500 gallons – 25 cents
- next 500 gallons – 20 cents
- any gallon above that – 15 cents

z = cost of x gallons

write $x = x_1 + x_2 + x_3$ where $0 \leq x_1, x_2 \leq 500$ and $x_3 \geq 0$

then $z = 25x_1 + 20x_2 + 15x_3$

$x_1 \geq 500y_1 \geq x_2 \geq 500y_2$

$My_2 \geq x_3 \qquad y_1, y_2 \in \{0,1\}$

Network problems - Travelling salesman

- find a shortest cycle through all nodes of $G = (V, E)$ where $V = \{1, \ldots, N\}$
- $x_{ij} = 1$ if edge (i, j) on the route
- $u_i = k$ if i is k-th node on the route

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$\sum_j x_{ij} = 1 \quad \text{and} \quad \sum_j x_{ji} = 1 \text{ for all } i \in V$$

$$u_i - u_j + Nx_{ij} \leq N - 1 \text{ for all } (i,j) \in E,\ i, j \neq 1$$

$$u_i \geq 0 \quad \text{and} \quad x_{ij} \in \{0,1\}$$

Why this works? The first two constraints make sure that exactly one edge is coming in and one edge going out of each node. So the chosen edges form a collection of cycles. If edge (i, j) is used, i.e. if $x_{ij} = 1$, then we have $u_i - u_j + N \leq N - 1$ and so $u_i < u_j$. This excludes any cycle on nodes $\{2, \ldots, N\}$. So there can only be one cycle, through all nodes. The objective function selects such a cycle of minimum cost.

Scheduling problems - non pre-emptive job scheduling

- schedule jobs on a machine(s), jobs have arrival time and processing time, minimize waiting time
- $x_{ijk} = 1$ if job j is i-th to be processed on machine k
- once job is started, it cannot be interrupted (no pre-emption)

Tricks

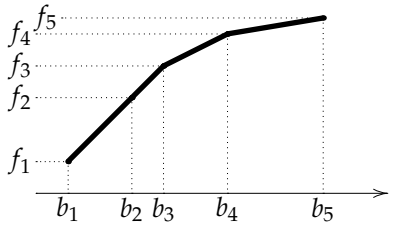
Or constraint: $f \leq 0$ or $g \leq 0$ $\iff$ 2 constraints $f \leq My$, $g \leq M(1-y)$ $\quad y \in \{0,1\}$ and M is a large number

If-then: if $f > 0$, then $g \geq 0$ $\iff$ 2 constraints $-g \leq My$, $f \leq M(1-y)$ $\quad y \in \{0,1\}$ and M is a large number

Piece-wise linear function z

- breakpoints $b_1, b_2, \ldots, b_t$
- with values $f_1, f_2, \ldots, f_t$

$$x = b_1x_1 + b_2x_2 + \ldots + b_tx_t$$
$$z = f_1x_1 + f_2x_2 + \ldots + f_tx_t$$
$$x_1 + x_2 + \ldots + x_t = 1$$



$$\begin{aligned} y_1 + y_2 + \ldots + y_{t-1} &= 1 \\ y_1, \ldots, y_t &\in \{0,1\} \\ x_1 &\leq y_1 \\ x_2 &\leq y_1 + y_2 \\ x_3 &\leq y_2 + y_3 \\ &\vdots \\ x_{t-1} &\leq y_{t-2} + y_{t-1} \\ x_t &\leq y_{t-1} \end{aligned}$$

Why this works? The variable y_i indicates that x is between b_i and b_{i+1}; if $y_i = 1$, then only x_i and x_{i+1} can be positive; so we have $x_i + x_{i+1} = 1$ and thus $x = b_ix_i + b_{i+1}(1 - x_i)$ and $z = f_ix_i + f_{i+1}(1 - x_i)$ where $0 \leq x_i \leq 1$; this exactly corresponds to the segment between (b_i, f_i) and (b_{i+1}, f_{i+1})

12.2 Cutting Planes

Let us go back to our toy factory problem. The toys are sold to a wholesale distributor who demands shipments in packs (boxes) of 30. There's no limit on how much we can produce.

$$\begin{array}{rrrrl}\text{Max} & 3x_1 & + & 2x_2 & \\ & x_1 & + & x_2 & \leq 80 \\ & 2x_1 & + & x_2 & \leq 100 \\ & x_1, x_2 & & & \geq 0\end{array} \quad \rightarrow \quad \begin{array}{l} 30w_1 = x_1 \\ 30w_2 = x_2 \\ w_1, w_2 \geq 0 \\ \text{integer}\end{array} \quad \rightarrow \quad \begin{array}{rrrrl}\text{Max} & 90w_1 & + & 60w_2 & \\ & 30w_1 & + & 30w_2 & \leq 80 \\ & 60w_1 & + & 30w_2 & \leq 100 \\ & w_1, w_2 & & & \geq 0 \\ & \text{integer} & & & \end{array}$$

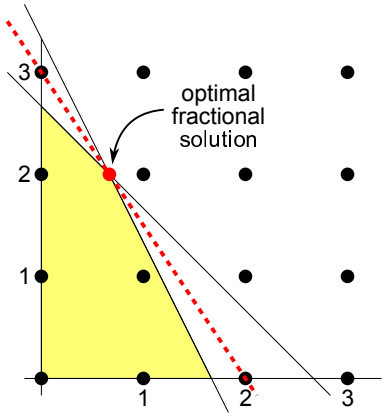
Consider the first constraint $30w_1 + 30w_2 \leq 80$. Let's divide both sides by 30. We obtain $w_1 + w_2 \leq \frac{8}{3}$. Notice that the left-hand side of the constraint is an integer, while the right-hand side is a fraction. Therefore

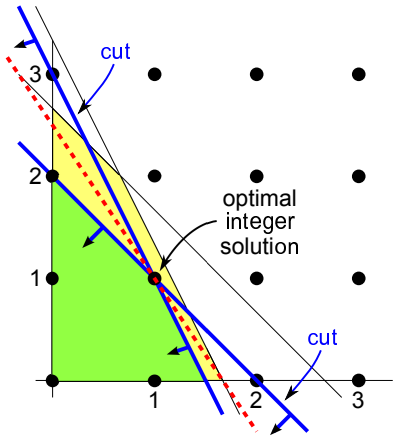
$$\underbrace{w_1 + w_2}_{\text{integer}} \leq \frac{8}{3} = 2 + \frac{2}{3} \quad \rightarrow \quad \underbrace{w_1 + w_2 - 2}_{\text{integer}} \leq \underbrace{\frac{2}{3}}_{<1} \quad \rightarrow \quad \underbrace{w_1 + w_2 - 2 \leq 0}_{\text{cut}}$$

What happens? We rewrite the constraint so that the left-hand side is integer and the right-hand side is a positive fraction less than one. Then, since the left-hand side is integer and is less than a fraction of one, it must be zero or less. This gives us a *new constraint* that we call a **cut** or a **cutting plane**. We can add this cut as our new constraint **without** removing any **integer** feasible solution.

Doing the same for the second constraint $60w_1 + 30w_2 \leq 100$:

$$60w_1 + 30w_2 \leq 100 \quad \rightarrow \quad \underbrace{2w_1 + w_2}_{\text{integer}} \leq \frac{10}{3} = 3 + \frac{1}{3} \quad \rightarrow \quad \underbrace{2w_1 + w_2 - 3}_{\text{integer}} \leq \underbrace{\frac{1}{3}}_{<1} \quad \rightarrow \quad \underbrace{2w_1 + w_2 - 3 \leq 0}_{\text{cut}}$$





Integer Linear Program

$$\begin{array}{rrrrl}\text{Max} & 90w_1 & + & 60w_2 & \\ & w_1 & + & w_2 & \leq 2 \\ & 2w_1 & + & w_2 & \leq 3 \\ & w_1, w_2 & & & \geq 0 \\ & \text{integer} & & & \end{array}$$

LP relaxation

$$\begin{array}{rrrrl}\text{Max} & 90w_1 & + & 60w_2 & \\ & w_1 & + & w_2 & \leq 2 \\ & 2w_1 & + & w_2 & \leq 3 \\ & w_1, w_2 & & & \geq 0\end{array}$$

Note: Every feasible solution to an integer linear program is also a feasible solution to its LP relaxation. Thus if the **LP relaxation has integer optimal solution**, then this is an **optimal solution** to the **ILP**.

The LP relaxation has an optimal solution $w_1 = w_2 = 1$ of value $z = 150$. Since this is an integer solution, it is also an optimal solution to the original integer linear program.

Conclusion: for optimal production, we should produce 1 box of each of the two toys.

Now, suppose that a different distributor demands that toy soldiers are delivered in boxes of 20, while toy trains in boxes of 25. We simplify the constrains by dividing both sides by 5 (and the objective by 10).

$$\begin{array}{rrrrl}\text{Max} & 3x_1 & + & 2x_2 & \\ & x_1 & + & x_2 & \leq 80 \\ & 2x_1 & + & x_2 & \leq 100 \\ & x_1, x_2 & & & \geq 0\end{array} \quad \begin{array}{l} 20w_1 = x_1 \\ 25w_2 = x_2 \\ w_1, w_2 \geq 0 \\ \text{integer}\end{array} \quad \begin{array}{rrrrl}\text{Max} & 60w_1 & + & 50w_2 & \\ & 20w_1 & + & 25w_2 & \leq 80 \\ & 40w_1 & + & 25w_2 & \leq 100 \\ & w_1, w_2 & & & \geq 0 \\ & \text{integer} & & & \end{array} \quad \begin{array}{rrrrl}\text{Max} & 6w_1 & + & 5w_2 & \\ & 4w_1 & + & 5w_2 & \leq 16 \\ & 8w_1 & + & 5w_2 & \leq 20 \\ & w_1, w_2 & & & \geq 0 \\ & \text{integer} & & & \end{array}$$

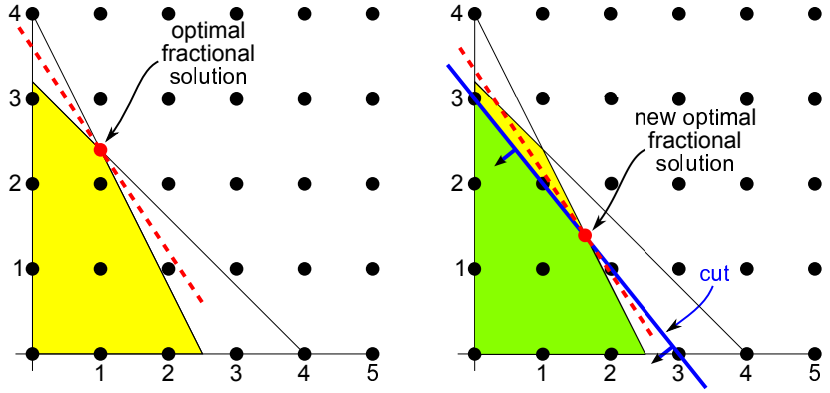
Notice that this time we cannot directly get fractions on the right-hand side while keeping the left-hand side integer.

However, let us multiply the first constraint by 3 and add it to the second constraint.

$$20w_1 + 20w_2 = 3(\underbrace{4w_1 + 5w_2}_{\leq 16}) + (\underbrace{8w_1 + 5w_2}_{\leq 20}) \leq 3 \times 16 + 20 = 68$$

We can now divide both sides by 20 to obtain $w_1 + w_2 \leq 3.4$. So again we can derive a cut.

$$\underbrace{w_1 + w_2}_{\text{integer}} \leq 3.4 = 3 + 0.4 \quad \rightarrow \quad \underbrace{w_1 + w_2 - 3}_{\text{integer}} \leq \underbrace{0.4}_{<1} \quad \rightarrow \quad \underbrace{w_1 + w_2 - 3 \leq 0}_{\text{cut}}$$



$$\begin{aligned} \text{Max } & 6w_1 + 5w_2 \\ & 4w_1 + 5w_2 \leq 16 \\ & 8w_1 + 5w_2 \leq 20 \\ & w_1 + w_2 \leq 3 \\ & w_1, w_2 \geq 0 \\ & \text{integer} \end{aligned}$$

We need a more systematic approach to finding cuts.

Gomory cuts

We can derive cuts directly from solving the LP relaxation. If the relaxation has an integer optimal solution, then we are done since this is also an optimal solution to the integer problem. If the optimal solution is **fractional**, we will use **rounding** to obtain a cut (as explained below). This cut will **cut off** the fractional solution and thus make the feasible region **smaller**. This makes sure that we **improve** in each step.

Consider the optimal dictionary to our problem.

$$\begin{aligned} \text{Max } & 6w_1 + 5w_2 \\ & 4w_1 + 5w_2 \leq 16 \\ & 8w_1 + 5w_2 \leq 20 \\ & w_1 + w_2 \leq 3 \\ & w_1, w_2 \geq 0 \\ & \text{integer} \end{aligned} \qquad \text{add slack variables (non-negative integers)}$$

$$\begin{aligned} \text{Max } & 6w_1 + 5w_2 \\ & 4w_1 + 5w_2 + x_3 = 16 \\ & 8w_1 + 5w_2 + x_4 = 20 \\ & w_1 + w_2 + x_5 = 3 \\ & w_1, w_2, x_3, x_4, x_5 \geq 0 \\ & \text{integer} \end{aligned}$$

$$\begin{aligned} w_1 &= \frac{5}{3} - \frac{1}{3}x_4 + \frac{5}{3}x_5 \\ w_2 &= \frac{4}{3} + \frac{1}{3}x_4 - \frac{8}{3}x_5 \\ x_3 &= \frac{8}{3} - \frac{1}{3}x_4 + \frac{20}{3}x_5 \\ \hline z &= \frac{50}{3} - \frac{1}{3}x_4 - \frac{10}{3}x_5 \end{aligned}$$

Choose any line of the dictionary with a **fractional** constant (in this case every line qualifies, including z). This choice is **arbitrary** but good practice suggest to choose the line where the fraction is closest to 1/2.

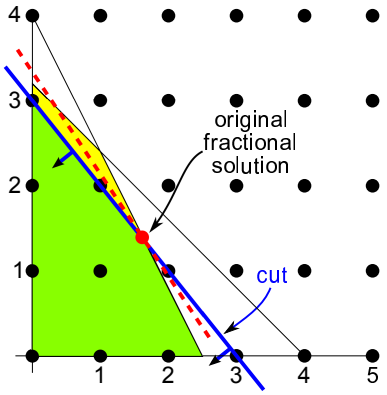
We pick the expression for z and move whole parts to the left, keeping only fractions on the right. This time we also have fractional coefficient with variables. We rearrange the expression so that on the right each variable has **negative** fractional coefficient, while the absolute constant is a **positive** fraction.

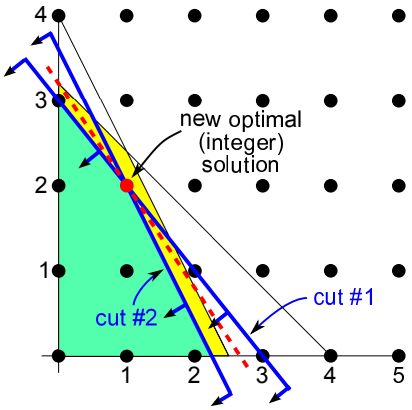
$$z = \underbrace{\tfrac{50}{3}}_{16+\frac{2}{3}} - \tfrac{1}{3}x_4 - \underbrace{\tfrac{10}{3}}_{-3-\frac{1}{3}} x_5 \quad \rightarrow \quad z + 3x_5 - 16 = \tfrac{2}{3} \underbrace{-\tfrac{1}{3}x_4 - \tfrac{1}{3}x_5}_{\text{negative fractions}} \quad \rightarrow \quad \underbrace{z + 3x_5 - 16 \leq 0}_{\text{cut}}$$

What happens? Because the coefficients of variables on the right-hand side are negative, the value of the right-hand side is 2/3 or less (cannot be more, since the variables are non-negative). So the value of the left-hand side is also at

most 2/3, but since the left-hand side is integer, the value is actually at most 0. We add the cut to our dictionary by introducing a **slack** (non-negative integer) and express it using the fractions.

$z + 3x_5 - 16 + x_6 = 0 \quad \rightarrow \quad x_6 = -(z + 3x_5 - 16) = -(\frac{2}{3} - \frac{1}{3}x_4 - \frac{1}{3}x_5) \quad \rightarrow \quad x_6 = -\frac{2}{3} + \frac{1}{3}x_4 + \frac{1}{3}x_5$





$$\begin{aligned} w_1 &= \tfrac{5}{3} - \tfrac{1}{3}x_4 + \tfrac{5}{3}x_5 \\ w_2 &= \tfrac{4}{3} + \tfrac{1}{3}x_4 - \tfrac{8}{3}x_5 \\ x_3 &= \tfrac{8}{3} - \tfrac{1}{3}x_4 + \tfrac{20}{3}x_5 \\ x_6 &= -\tfrac{2}{3} + \tfrac{1}{3}x_4 + \tfrac{1}{3}x_5 \\ \hline z &= \tfrac{50}{3} - \tfrac{1}{3}x_4 - \tfrac{10}{3}x_5 \end{aligned}$$

Dually feasible dictionary
$\rightarrow$ use Dual Simplex

x_6 leaves (value < 0)

ratio test:
$x_4 : (1/3)/(1/3) = 1$
$x_5 : (10/3)/(1/3) = 10$

$\rightarrow x_4$ enters

$$\begin{aligned} w_1 &= 1 + 2x_5 - x_6 \\ w_2 &= 2 - 3x_5 + x_6 \\ x_3 &= 2 + 7x_5 - x_6 \\ x_4 &= 2 - x_5 + 3x_6 \\ \hline z &= 16 - 3x_5 - x_6 \end{aligned}$$

Optimal integer solution found $w_1 = 1$, $w_2 = 2$ with value $z = 16$ (corresponds to \$160 of profit).

Conclusion: optimal production consists of producing one pack of toy soldiers, and two packs of toy trains.

Cutting planes algorithm

This works **only** for **pure** integer linear programs.

1. Solve the LP relaxation.
2. If the LP is **infeasible**, then **report** that the problem is **infeasible**, and **stop**.
3. If all variables are integers, then **report** the solution and **stop**.
4. Else in the optimal dictionary, pick any line with fractional constant.
   (a) Rewrite the line by moving whole parts to the left so that
   - the absolute constant on the right is a positive fraction (less than 1),
   - the coefficients of variables on the right are negative fractions (less than 1 in absolute value).

   (b) Make the **right-hand side** ≤ 0 to form a new constraint (cut).
   (c) Introduce a slack variable to the cut and add the resulting equation to the dictionary.
   (d) Solve the resulting LP using the Dual Simplex method and then go back to 2.

Example. suppose that the dictionary contains

$$x_1 = \tfrac{5}{2} - \tfrac{3}{2}x_2 + \tfrac{7}{2}x_3 + 2x_4$$

Rewrite by moving whole parts to the left so that the coefficient of variables on the right are **negative**.

$$x_1 = \underbrace{\tfrac{5}{2}}_{2+\frac{1}{2}} \underbrace{- \tfrac{3}{2}}_{-1-\frac{3}{2}} x_2 \underbrace{+ \tfrac{7}{2}}_{4-\frac{1}{2}} x_3 + 2x_4$$

$$\boxed{x_1 + x_2 - 4x_3 - 2x_4 - 2 = \tfrac{1}{2} - \tfrac{1}{2}x_2 - \tfrac{1}{2}x_3}$$

(Notice: x_3 had positive coefficient $+\frac{7}{2}$, and we had to take $\frac{7}{2} = 4 - \frac{1}{2}$ to get a negative coeff on the right.)
The right-hand side is < 1 but must be integer (since lhs is). We introduce the cut that makes is at most 0.

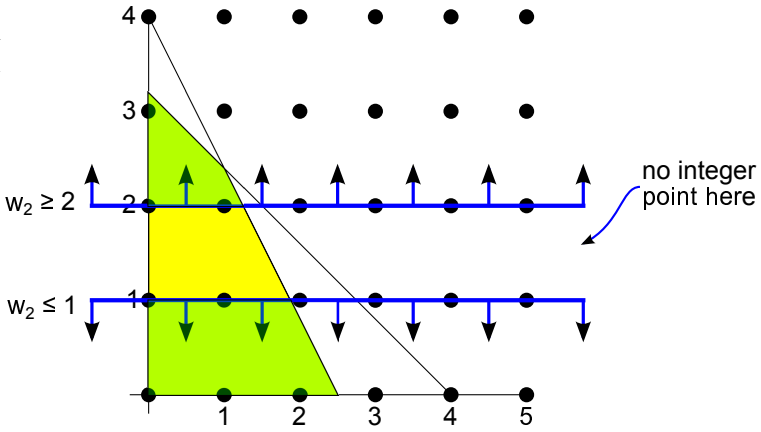
$$\tfrac{1}{2} - \tfrac{1}{2}x_2 - \tfrac{1}{2}x_3 \leq 0 \qquad \text{add slack} \rightarrow \text{new constraint} \qquad \boxed{x_5 = -\tfrac{1}{2} + \tfrac{1}{2}x_2 + \tfrac{1}{2}x_3}$$

12.3 Branch and Bound

For illustration, consider the toy factory problem where toy soldiers are delivered in boxes of 20 and toy trains in boxes of 25.

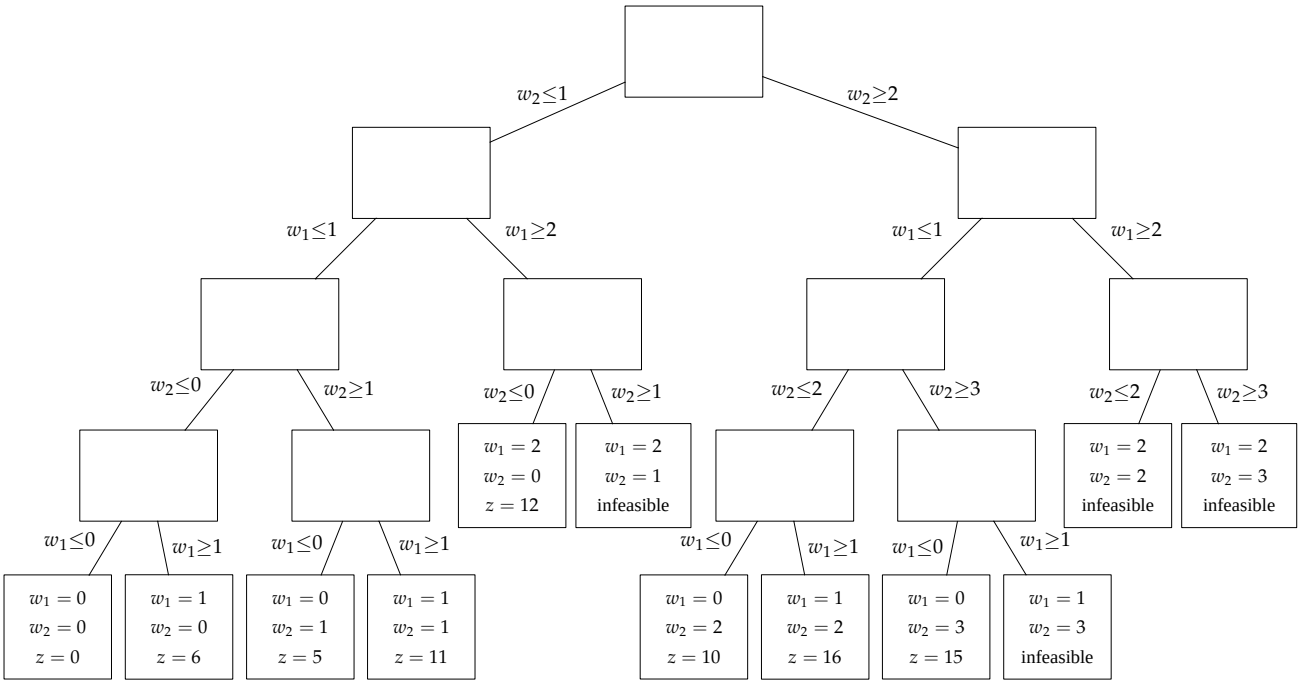
$$\begin{aligned} \text{Max } & 6w_1 + 5w_2 \\ & 4w_1 + 5w_2 \leq 16 \\ & 8w_1 + 5w_2 \leq 20 \\ & 0 \leq w_1 \leq 2 \\ & 0 \leq w_2 \leq 3 \\ & w_1, w_2 \text{ integer} \end{aligned}$$

Notice that we also assume that the variables w_1 and w_2 are upper-bounded (by 2 resp. 3).

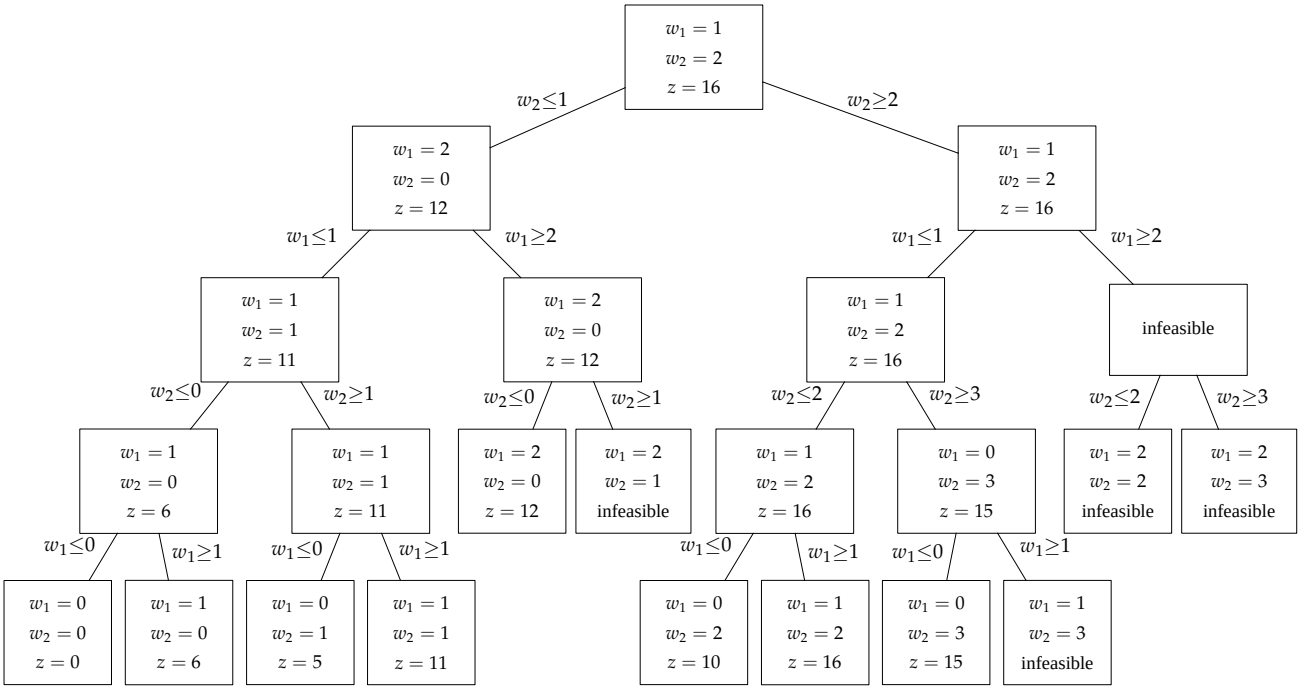


Enumerating solutions by Branching

Consider the variable w_2. Since we are only interested solutions where this variable is an integer, we can also assume that w_2 does not attain any value strictly between 1 and 2. This splits the feasible region onto two parts, one where $w_2 \leq 1$ and one where $w_2 \geq 2$. We solve the problem on each of the two parts independently and pick the better of the two solutions. In other words, we **branch** into two **subcases**. In each of the subcases, we can again consider an integer variable, say w_1, and exclude all fractional values between, say zero and one. We branch into two further subcases and continue this process until the values of all integer variables have been determined. We can visualize this using a **branching tree**.



We evaluate the constraints in the leaves and the propagate best solution bottom-up.



Optimal solution is $w_1 = 1$, $w_2 = 2$ with $z = 16$.

Using bounds in branching

We can shorten the branching process if we have some **heuristics**, a way to **bound** or **estimate** the value of the **objective** function in subsequent subcases. For instance, in the above, if we somehow know that best possible value in the subcase $w_2 \leq 1$ is at most 12, and we also know that $w_1 = 0$, $w_2 = 3$ is a solution of value 15, then we don't need to branch into this subcase; the best solution that we would find there would be worse that the one we already know. This idea is at the basis of the Branch-and-Bound method.

How do we find a bound on the objective? We solve the LP relaxation by some means:

1. Graphical method
2. Simplex algorithm
3. Special purpose algorithms, for instance:
   (a) Knapsack - Fractional Knapsack
   (b) TSP - Assignment problem
   (c) Facility location - Network algorithms

Branch and bound using the Simplex algorithm

The process goes as follows: we **solve** the **LP relaxation** of the problem. If the optimal solution is such that all variables have **integer** values, then we have found an **integer solution** and we are done. Otherwise, there is at least one variable whose value is **fractional**. We use this variable to branch. Namely, suppose that the variable is x_i and the value is $x_i = a$. We branch into **two cases**: (1) $x_i \leq \lfloor a \rfloor$, and (2) $x_i \geq \lceil a \rceil$. Note that this shrinks the feasible region, since it cuts-off the solution that we used to branch (in that solution, x_i had value a, but in neither of the two subcases x_i can have this value). This way we always make progress and only branch close to the fractional optimum, which is where the integer optimum is likely to be (if exists).

The LP relaxation can be solved by the Graphical method (for 2-dimensional problems). For more general problems,

we can use the Simplex algorithm and **reuse** dictionaries in subsequent steps.

Integer program

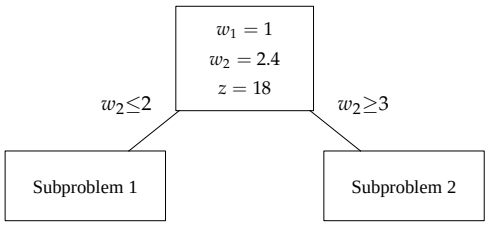
$$\begin{aligned} \text{Max } & 6w_1 + 5w_2 \\ & 4w_1 + 5w_2 \le 16 \\ & 8w_1 + 5w_2 \le 20 \\ & w_1, w_2 \ge 0 \\ & \text{integer} \end{aligned}$$

$$\begin{aligned} \text{Max } & 6w_1 + 5w_2 \\ & 4w_1 + 5w_2 + x_3 = 16 \\ & 8w_1 + 5w_2 + x_4 = 20 \\ & w_1, w_2 \ge 0 \qquad x_3, x_4 \ge 0 \\ & \text{integer} \end{aligned}$$

$$\begin{aligned} w_1 &= 1 + 0.25x_3 - 0.25x_4 \\ w_2 &= 2.4 - 0.4x_3 + 0.2x_4 \\ \hline z &= 18 - 0.5x_3 - 0.5x_4 \end{aligned}$$

Optimal fractional solution

The optimal solution is fractional, because $w_2 = 2.4$ is not an integer. We branch into $w_2 \le 2$ and $w_2 \ge 3$.



Subproblem 1:

$$\begin{aligned} w_1 &= 1 + 0.25x_3 - 0.25x_4 \\ w_2 &= 2.4 - 0.4x_3 + 0.2x_4 \\ \hline z &= 18 - 0.5x_3 - 0.5x_4 \end{aligned}$$

Bounds:
$0 \le w_1 \le \infty$
$0 \le w_2 \le 2$

Subproblem 2:

$$\begin{aligned} w_1 &= 1 + 0.25x_3 - 0.25x_4 \\ w_2 &= 2.4 - 0.4x_3 + 0.2x_4 \\ \hline z &= 18 - 0.5x_3 - 0.5x_4 \end{aligned}$$

Bounds:
$0 \le w_1 \le \infty$
$3 \le w_2 \le \infty$

In both subproblems the dictionary is not feasible. To fix it, we use the Upper-bounded Dual Simplex method. Let us first solve **Subproblem 1**.

$$\begin{aligned} w_1 &= 1 + 0.25x_3 - 0.25x_4 \\ w_2 &= 2.4 - 0.4x_3 + 0.2x_4 \\ \hline z &= 18 - 0.5x_3 - 0.5x_4 \end{aligned}$$

Bounds:
$0 \le w_1 \le \infty$
$0 \le w_2 \le 2$

check lower bounds: $0 \le 1 = w_1$, $0 \le 2.4 = w_2$
check upper bounds: $w_1 = 1 \le \infty$, $w_2 = 2.4 \not\le 2$
$\rightarrow$ replace w_2 by $w_2 = 2 - w_2'$

$$\begin{aligned} w_1 &= 1 + 0.25x_3 - 0.25x_4 \\ w_2' &= -0.4 + 0.4x_3 - 0.2x_4 \\ \hline z &= 18 - 0.5x_3 - 0.5x_4 \end{aligned}$$

Bounds:
$0 \le w_1 \le \infty$
$0 \le w_2 \le 2$

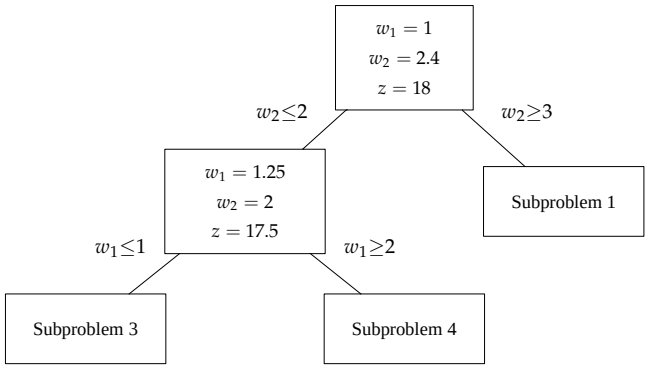
check lower bounds: $0 \le 1 = w_1$, $0 \not\le -0.4 = w_2'$
$\rightarrow w_2'$ leaves, ratio test $x_3 : 0.5/0.4 = 1.25$
x_4 : *no constraint*
$\rightarrow x_3$ enters

$$\begin{aligned} w_1 &= 1.25 - 0.125x_4 + 0.625w_2' \\ x_3 &= 1 + 0.5x_4 + 2.5w_2' \\ \hline z &= 17.5 - 0.75x_4 - 1.25w_2' \end{aligned}$$

Bounds:
$0 \le w_1 \le \infty$
$0 \le w_2 \le 2$

optimal solution found:
$w_1 = 1.25$, $w_2' = 0 \rightarrow w_2 = 2$
$z = 17.5$

The optimal solution is fractional, since $w_1 = 1.25$. We branch on $w_1 \le 1$ and $w_1 \ge 2$.



Subproblem 3:

$$\begin{aligned} w_1 &= 1.25 - 0.125x_4 + 0.625w_2' \\ x_3 &= 1 + 0.5x_4 + 2.5w_2' \\ \hline z &= 17.5 - 0.75x_4 - 1.25w_2' \end{aligned}$$

Bounds:
$0 \le w_1 \le 1$
$0 \le w_2 \le 2$

Subproblem 4:

$$\begin{aligned} w_1 &= 1.25 - 0.125x_4 + 0.625w_2' \\ x_3 &= 1 + 0.5x_4 + 2.5w_2' \\ \hline z &= 17.5 - 0.75x_4 - 1.25w_2' \end{aligned}$$

Bounds:
$2 \le w_1 \le \infty$
$0 \le w_2 \le 2$

We again solve **Subproblem 3** using the Upper-bounded Dual Simplex method.

$$\begin{array}{rcrcr} w_1 &=& 1.25 &-& 0.125x_4 &+& 0.625w_2' \\ x_3 &=& 1 &+& 0.5x_4 &+& 2.5w_2' \\ \hline z &=& 17.5 &-& 0.75x_4 &-& 1.25w_2' \end{array}$$

Bounds: $0 \le w_1 \le 1$, $0 \le w_2 \le 2$

check lower bounds: $0 \le 1.25 = w_1, 0 \le 1 = x_3$
check upper bounds: $w_1 = 1.25 \not\le 1$, $x_3 = 1 \le \infty$

$\rightarrow$ replace w_1 by $w_1 = 1 - w_1'$

$$\begin{array}{rcrcr} w_1' &=& -0.25 &+& 0.125x_4 &-& 0.625w_2' \\ x_3 &=& 1 &+& 0.5x_4 &+& 2.5w_2' \\ \hline z &=& 17.5 &-& 0.75x_4 &-& 1.25w_2' \end{array}$$

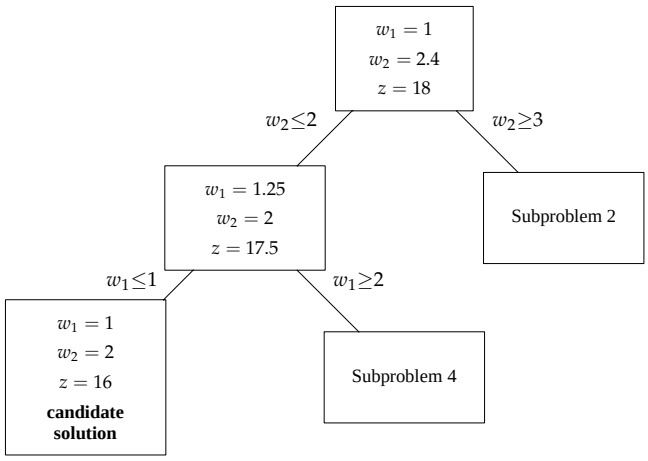
Bounds: $0 \le w_1 \le 1$, $0 \le w_2 \le 2$

check lower bounds: $0 \not\le -0.25 = w_1', 0 \le 1 = x_3$
$\rightarrow w_1'$ leaves, ratio test $x_4 : 0.75/0.125 = 6$
w_2' : *no constraint*
$\rightarrow x_4$ enters

$$\begin{array}{rcrcr} x_3 &=& 2 &+& 4w_1' &+& 5w_2' \\ x_4 &=& 2 &+& 8w_1' &+& 5w_2' \\ \hline z &=& 16 &-& 6w_1' &-& 5w_2' \end{array}$$

Bounds: $0 \le w_1 \le \infty$, $0 \le w_2 \le 2$

optimal (integer) solution found:
$w_1' = 0 \rightarrow w_1 = 1, \quad w_2' = 0 \rightarrow w_2 = 2$
$z = 16 \quad \rightarrow$ **candidate** integer solution



Now we go back and solve **Subproblem 4**. Since $w_1 \ge 2$ we substitute: $w_1 = 2 + w_3$ where $w_3 \ge 0$ is integer.

$$\begin{array}{rcrcr} w_3 &=& -0.75 &-& 0.125x_4 &+& 0.625w_2' \\ x_3 &=& 1 &+& 0.5x_4 &+& 2.5w_2' \\ \hline z &=& 17.5 &-& 0.75x_4 &-& 1.25w_2' \end{array}$$

Bounds: $0 \le w_3 \le \infty$, $0 \le w_2 \le 2$

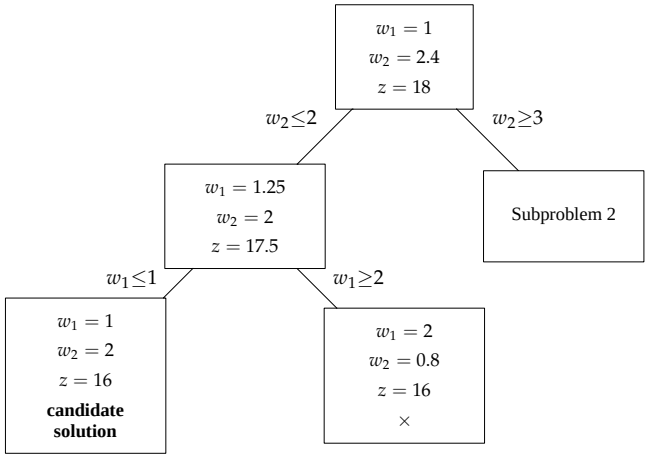
check lower bounds: $0 \not\le -0.75 = w_3, 0 \le 1 = x_3$
$\rightarrow w_3$ leaves, ratio test x_4 : *no constraint*
w_2' : $1.25/0.625 = 2$
$\rightarrow w_2'$ enters

$$\begin{array}{rcrcr} w_2' &=& 1.2 &+& 0.2x_4 &+& 1.6w_3 \\ x_3 &=& 4 &+& x_4 &+& 4w_3 \\ \hline z &=& 16 &-& x_4 &-& 2w_3 \end{array}$$

Bounds: $0 \le w_1 \le \infty$, $0 \le w_2 \le 2$

optimal solution found:
$w_3 = 0 \rightarrow w_1 = 2, \quad w_2' = 1.2 \rightarrow w_2 = 0.8$
$z = 16$

Optimal solution is fractional, but its value is $z = 16$. Thus we **do not branch**, since we already have a **candidate integer solution** of value $z = 16$.



Now we go back and solve **Subproblem 2**. Since $w_2 \geq 3$, we substitute $w_2 = 3 + w_4$ where $w_4 \geq 0$ is integer.

$$\begin{array}{rcrcrcl} w_1 & = & 1 & + & 0.25x_3 & - & 0.25x_4 \\ w_4 & = & -0.6 & - & 0.4x_3 & + & 0.2x_4 \\ \hline z & = & 18 & - & 0.5x_3 & - & 0.5x_4 \end{array}$$

Bounds:
$0 \leq w_1 \leq \infty$
$0 \leq w_4 \leq \infty$

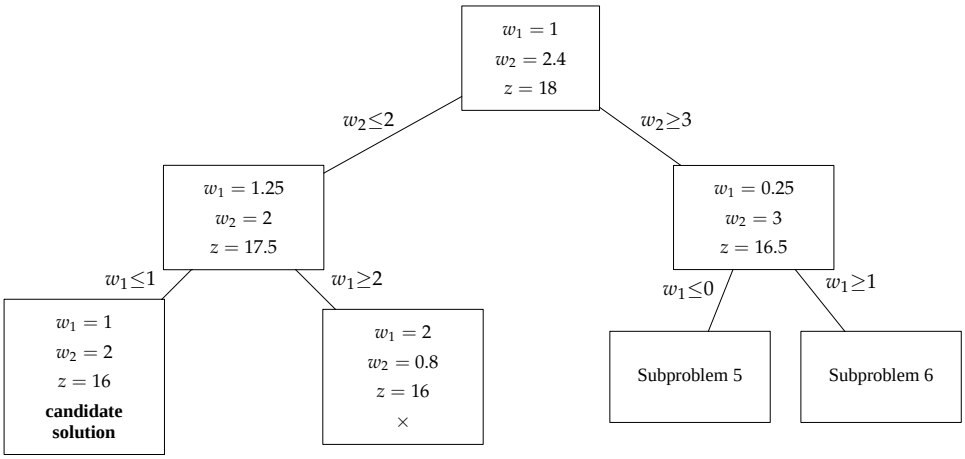
check lower bounds: $0 \leq 1 = w_1$, $0 \not\leq -0.6 = w_4$
$\rightarrow w_4$ leaves, ratio test x_3 : *no constraint*
$x_4 : 0.5/0.2 = 2.5$
$\rightarrow x_4$ enters

$$\begin{array}{rcrcrcl} w_1 & = & 0.25 & - & 0.25x_3 & - & 1.25w_4 \\ x_4 & = & 3 & + & 2x_3 & + & 5w_4 \\ \hline z & = & 16.5 & - & 1.5x_3 & - & 2.5w_4 \end{array}$$

Bounds:
$0 \leq w_1 \leq \infty$
$0 \leq w_4 \leq \infty$

optimal solution found:
$w_1 = 0.25$, $w_4 = 0 \rightarrow w_2 = 3$
$z = 16.5$

Optimal solution is fractional, but its value $z = 16.5$ is more than $z = 16$ of the candidate solution. It is still possible that a better solution can be found by branching (here we ignore the fact that optimal z should also be an integer, otherwise we would not need to branch). Since $w_1 = 0.25$, we branch on $w_1 \leq 0$ and $w_1 \geq 1$.



Subproblem 5:

$$\begin{array}{rcrcrcl} w_1 & = & 0.25 & - & 0.25x_3 & - & 1.25w_4 \\ x_4 & = & 3 & + & 2x_3 & + & 5w_4 \\ \hline z & = & 16.5 & - & 1.5x_3 & - & 2.5w_4 \end{array}$$

Bounds:
$0 \leq w_1 \leq 0$
$0 \leq w_4 \leq \infty$

Subproblem 6:

$$\begin{array}{rcrcrcl} w_1 & = & 0.25 & - & 0.25x_3 & - & 1.25w_4 \\ x_4 & = & 3 & + & 2x_3 & + & 5w_4 \\ \hline z & = & 16.5 & - & 1.5x_3 & - & 2.5w_4 \end{array}$$

Bounds:
$1 \leq w_1 \leq \infty$
$0 \leq w_4 \leq \infty$

Let us now solve **Subproblem 5**.

$$\begin{array}{rcrcrcl} w_1 & = & 0.25 & - & 0.25x_3 & - & 1.25w_4 \\ x_4 & = & 3 & + & 2x_3 & + & 5w_4 \\ \hline z & = & 16.5 & - & 1.5x_3 & - & 2.5w_4 \end{array}$$

Bounds:
$0 \leq w_1 \leq 0$
$0 \leq w_4 \leq \infty$

check lower bounds: $0 \leq 0.25 = w_1$, $0 \leq 3 = x_4$
check upper bounds: $w_1 = 0.25 \not\leq 0$, $x_4 = 3 \leq \infty$
$\rightarrow$ replace w_1 by $w_1 = 0 - w_1'$

$$\begin{array}{rcrcrcl} w_1' & = & -0.25 & + & 0.25x_3 & + & 1.25w_4 \\ x_4 & = & 3 & + & 2x_3 & + & 5w_4 \\ \hline z & = & 16.5 & - & 1.5x_3 & - & 2.5w_4 \end{array}$$

Bounds:
$0 \leq w_1 \leq 0$
$0 \leq w_4 \leq \infty$

check lower bounds: $0 \not\leq -0.25 = w_1'$, $0 \leq 3 = x_4$
$\rightarrow w_1'$ leaves, ratio test $x_3 : 1.5/0.25 = 6$
$w_4 : 2.5/1.25 = 2$
$\rightarrow w_4$ enters

$$\begin{array}{rcrcrcl} w_4 & = & 0.2 & - & 0.2x_3 & + & 0.8w_1' \\ x_4 & = & 4 & - & x_3 & + & 4w_1' \\ \hline z & = & 16 & - & x_3 & - & 2w_1' \end{array}$$

Bounds:
$0 \leq w_1 \leq 0$
$0 \leq w_4 \leq \infty$

optimal solution found:
$w_1' = 0 \rightarrow w_1 = 0$, $w_4 = 0.2 \rightarrow w_2 = 3.2$
$z = 16$

The solution has value $z = 16$ which is no better than our candidate solution. We do not branch.

We now solve **Subproblem 6**. Since $w_1 \geq 1$, we substitute $w_1 = 1 + w_5$, where $w_5 \geq 0$ and integer.

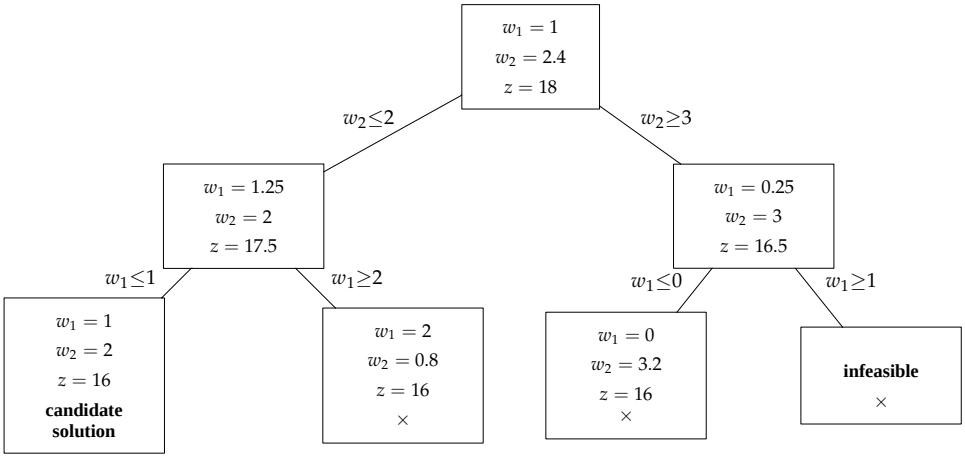
$$\begin{array}{rcrcrcl} w_5 & = & -0.75 & - & 0.25x_3 & - & 1.25w_4 \\ x_4 & = & 3 & + & 2x_3 & + & 5w_4 \\ \hline z & = & 16.5 & - & 1.5x_3 & - & 2.5w_4 \end{array}$$

Bounds:
$0 \leq w_5 \leq \infty$
$0 \leq w_4 \leq \infty$

check lower bounds: $0 \not\leq -0.75 = w_5$, $0 \leq 3 = x_4$
$\rightarrow w_5$ leaves, ratio test x_3 : *no constraint*
x_4 : *no constraint*
$\rightarrow$ no variable can enter $\rightarrow$ **infeasible** LP

Since the problem is infeasible, we do not need to branch anymore (no further restriction can make it feasible).

We now have solved all subproblems. We can thus summarize.



We conclude that there exist an integer feasible solution (our candidate solution), and it is also an optimal integer solution: $w_1 = 1$, $w_2 = 2$ of value $z = 16$. This concludes the algorithm.

Branch and bound for Knapsack

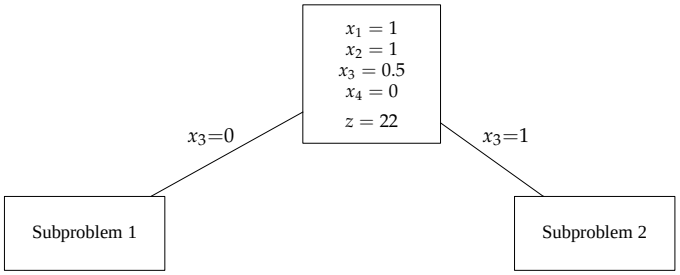
$$\begin{aligned} \text{Max } z \;=\; & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_1, x_2, x_3, x_4 \in \{0,1\} \\ & z \text{ is integer} \end{aligned}$$

LP relaxation is called Fractional Knapsack. Can be solved by calculating prices of items per unit of weight and choosing items of highest unit price first.

Item	1	2	3	4
Unit price	$\frac{8}{5} = 1.6$	$\frac{11}{7} = 1.57$	$\frac{6}{4} = 1.5$	$\frac{4}{3} = 1.33$

We pick $x_1 = 1$, since item 1 has highest unit cost. The remaining budget is $14 - 5 = 9$. Then we pick $x_2 = 1$, since item 2 has next highest unit cost. The budget reduces to $9 - 7 = 2$. Then we pick item 3 which has next highest unit cost, but we can only pick $x_3 = 0.5$ and use up all the budget.

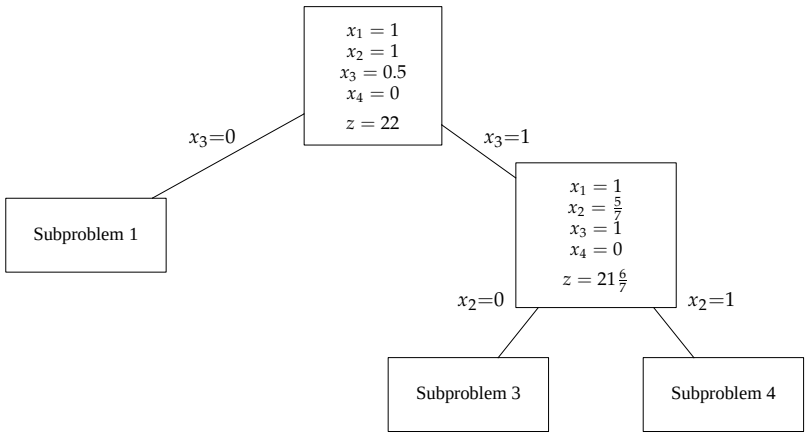
The total price is $z = 8 + 11 + 0.5 \times 6 = 22$. Since $x_3 = 0.5$, we branch on $x_3 = 0$ and $x_3 = 1$.



Subproblem 2: $x_3 = 1$

$$\begin{aligned} \text{Max } & 8x_1 + 11x_2 + 6 + 4x_4 \\ & 5x_1 + 7x_2 + 4 + 3x_4 \leq 14 \end{aligned} \quad \rightarrow \quad \begin{aligned} \text{Max } & 6 + 8x_1 + 11x_2 + 4x_4 \\ & 5x_1 + 7x_2 + 3x_4 \leq 10 \end{aligned}$$

Again we pick $x_1 = 1$ and budget reduces to $10 - 5 = 5$ so we can pick only $x_2 = 5/7$ from item 2. The total price is $z = 6 + 8 + 11 \times \frac{5}{7} = 21\frac{6}{7}$. We branch on $x_2 = 0$ and $x_2 = 1$.



Subproblem 3: $x_2 = 0$

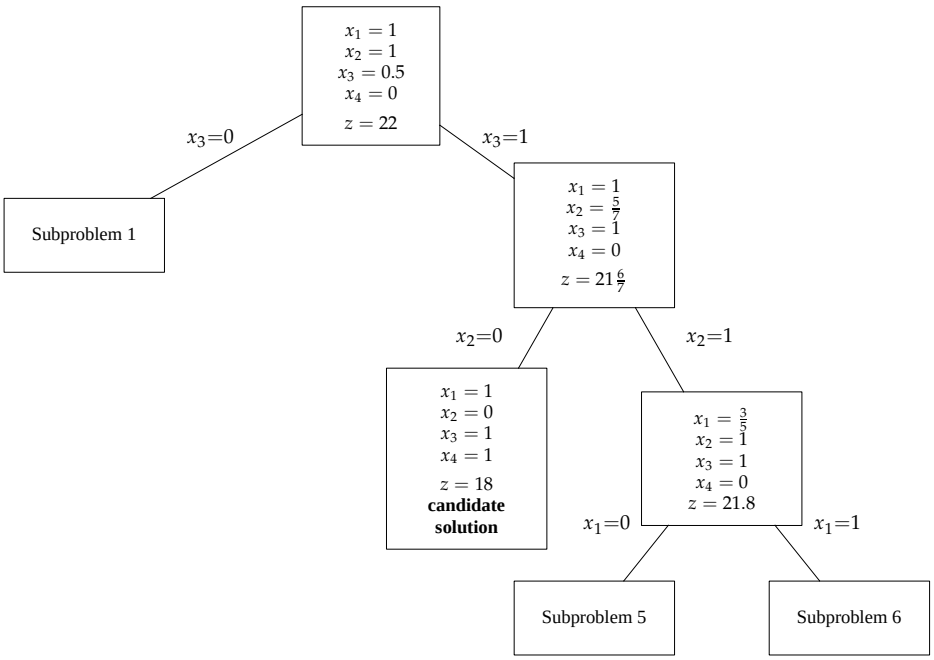
$$\begin{aligned} \text{Max } & 6 + 8x_1 + 4x_4 \\ & 5x_1 + 3x_4 \leq 10 \end{aligned}$$

pick $x_1 = 1$ and $x_4 = 1$ (budget is not used up completely)
the total price is $z = 6 + 8 + 4 = 18$ $\rightarrow$ **candidate solution**

Subproblem 4: $x_2 = 1$

$$\begin{aligned} \text{Max } & 17 + 8x_1 + 4x_4 \\ & 5x_1 + 3x_4 \leq 3 \end{aligned}$$

pick $x_1 = 3/5$ and use up the budget
the total price is $z = 17 + 8 \times \frac{3}{5} = 21.8$
$\rightarrow$ branch on $x_1 = 0$ and $x_1 = 1$



Subproblem 5: $x_1 = 0$

$$\begin{aligned} \text{Max } & 17 + 4x_4 \\ & 3x_4 \leq 3 \end{aligned}$$

pick $x_4 = 1$ and use up the budget
the total price is $z = 17 + 4 = 21$ $\rightarrow$ **candidate solution**

Subproblem 4: $x_1 = 1$

$$\begin{aligned} \text{Max } & 25 + 4x_4 \\ & 3x_4 \leq -2 \end{aligned}$$

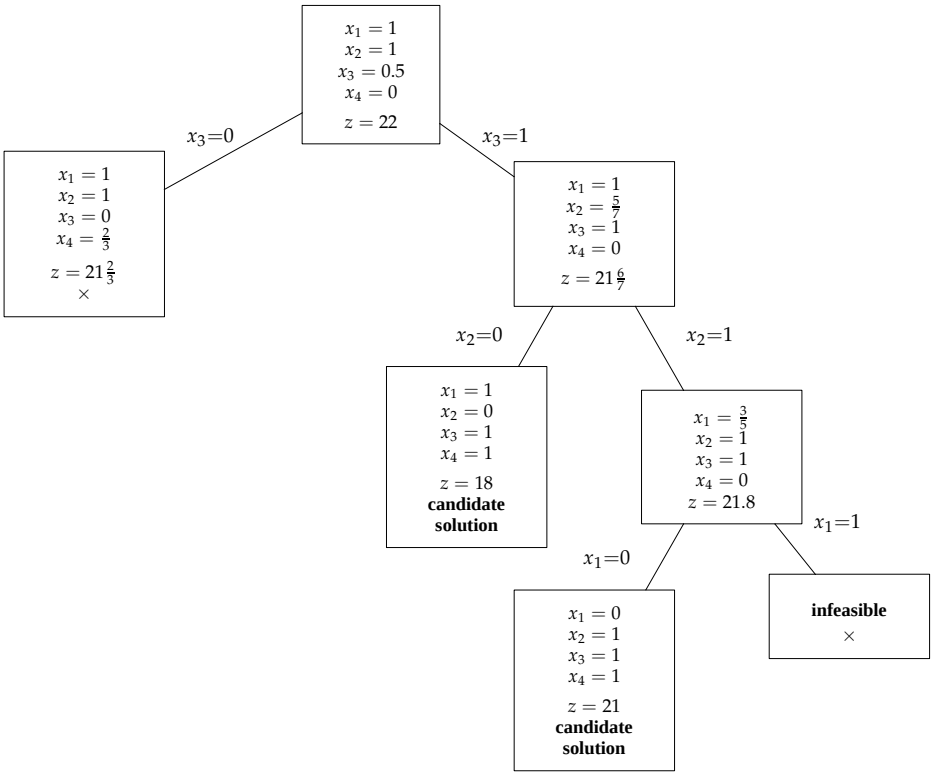
problem **infeasible**

Subproblem 1: $x_3 = 0$

$$\begin{array}{rrrrrrr} \text{Max} & 8x_1 & + & 11x_2 & + & 4x_4 & \\ & 5x_1 & + & 7x_2 & + & 3x_4 & \leq 14 \end{array}$$

pick $x_1 = 1$ then $x_2 = 1$ which reduces the budget to 2;
then we pick $x_4 = 2/3$ to use the up the budget.
The total price is $z = 8 + 11 + 4 \times \frac{2}{3} = 21\frac{2}{3}$

$\rightarrow$ we **do not branch**, since z must be integer and so best integer subproblem of Subproblem 1 has value $z \leq 21$ but we already have a candidate solution of value $z = 21$



Short comparison of Integer Programming Methods

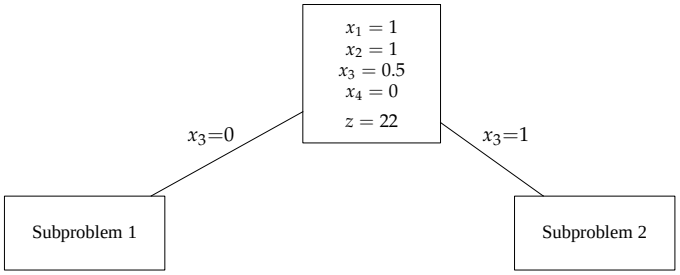
1. Cutting planes – adds a constrain at each step, does not branch, once an integer solution found we stop
2. Branch-and-bound – branches into (independent) subproblems, does not add constraints, only changes bounds on variables, if an integer solution is found, we cannot stop; all subproblems must be explored before we can declare that we found an optimal solution
3. Dynamic programming – building up a solution by reusing (dependent) subproblems; more efficient than Branch-and-Bound but only works for some problems (for instance, it works for the Knapsack Problem but not so much for the Traveling Salesman Problem)

13 Dynamic Programming

Knapsack

Consider the Knapsack problem from the previous lecture.

$$\begin{aligned} \text{Max } z \;=\; & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ & 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\ & x_1, x_2, x_3, x_4 \in \{0,1\} \\ & z \text{ is integer} \end{aligned}$$



Subproblem 1: $x_3 = 0$

$$\begin{aligned} \text{Max } & 8x_1 + 11x_2 + 4x_4 \\ & 5x_1 + 7x_2 + 3x_4 \le 14 \end{aligned}$$

Subproblem 2: $x_3 = 1$

$$\begin{aligned} \text{Max } & 6 + 8x_1 + 11x_2 + 4x_4 \\ & 5x_1 + 7x_2 + 3x_4 \le 10 \end{aligned}$$

Note that every subproblem is characterized by the set of remaining objects (variables) and the total **budget** (the right-hand side). Observe that the above two subproblems only differ in the value of budget (as we can ignore the absolute constant in the objective function). We should exploit this symmetry.

Instead of solving just for 10 and 14, we solve the subproblem for **all** meaningful values of the right-hand side (here for values $1, 2, \ldots, 14$). Having done that, we pick the best solution. This may seem wasteful but it can actually be faster. To make this work efficiently, we branch systematically on variables $x_1, x_2, \ldots$

Let us describe it first more generally. Consider the problem (all coefficients are integers)

$$\begin{aligned} \max\ & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ & d_1x_1 + d_2x_2 + \ldots + d_nx_n \le B \\ & x_1, \ldots, x_n \in \{0,1\} \end{aligned}$$

For every $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, B\}$, we solve a subproblem with variables $x_1, \ldots, x_i$ and budget j.

$$\begin{aligned} \max\ & c_1x_1 + c_2x_2 + \ldots + c_ix_i \\ & d_1x_1 + d_2x_2 + \ldots + d_ix_i \leq j \\ & x_1, \ldots, x_i \in \{0,1\} \end{aligned}$$

Let $f_i(j)$ denote the value of this solution. We want the value of $f_n(B)$.

How can we calculate $f_i(j)$? We observe that the optimal solution consisting of first i items either contains the i-th item or it does not. If it does not contain the i-th item, then the value of $f_i(j)$ is the same as that of $f_{i-1}(j)$; the best solution using just the first $i-1$ items. If, on the other hand, an optimal solution using the first i items contains the i-th item, then removing this item from the solution gives an **optimal solution** for **first** $i-1$ items **with budget** $j-d_i$. (Convince yourself of this fact.) So the value of $f_i(j)$ is obtained by taking $f_{i-1}(j-d_i)$ and adding the value c_i of item i. We don't know which of the two situations happens, but by taking the better of the two, we always choose correctly.

> What this shows is that from optimal solutions to smaller subproblems we can build an optimal solution to a larger subproblem. We say that the problem has **optimal substructure**.

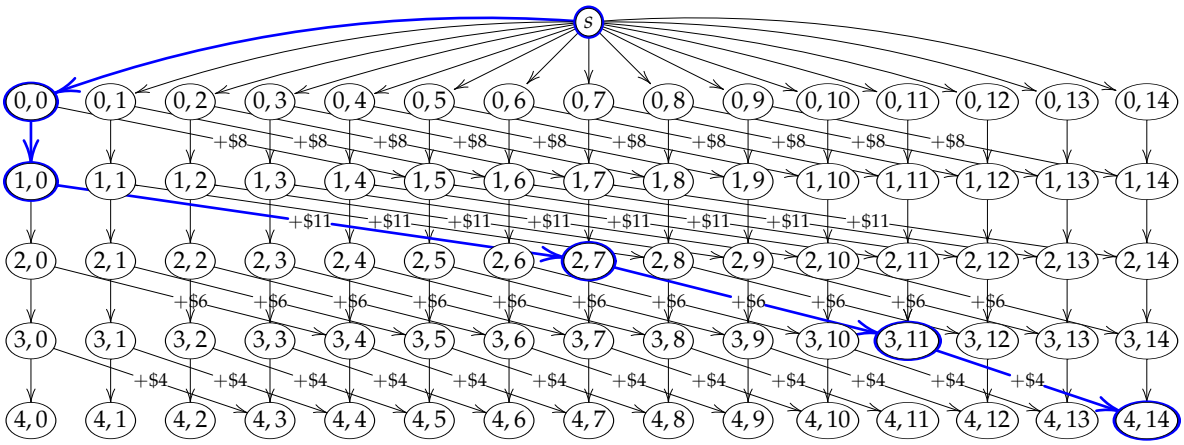
As we just described, the function $f_i(j)$ satisfies the following **recursion**:

$$f_i(j) = \begin{cases} 0 & i = 0 \\ \max\{f_{i-1}(j),\ c_i + f_{i-1}(j-d_i)\} & i \geq 1 \end{cases}$$

We can calculate it by filling the table of all possible values.

$f_i(j)$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	8	8	8	8	8	8	8	8	8	8
2	0	0	0	0	0	8	8	11	11	11	11	11	19	19	19
3	0	0	0	0	6	8	8	11	11	14	14	17	19	19	19
4	0	0	0	4	6	8	8	11	12	14	15	17	19	19	21

+$6



(edges with no cost indicated have \$0 cost)

The longest path (in **bold**) from s to $t = (4,14)$ gives the optimal solution.

Dynamic programming characteristics

- Problem can be divided into stages
- Each stage has a (finite) number of possible **states**, each associated a value.
- Next stage only depends on the values of states of the previous stage

Resource allocation

Knapsack with fixed costs.

Investment 1: investing $x_1 > 0$ dollars yields $c_1(x_1) = 7x_1 + 2$ dollars, where $c_1(0) = 0$
Investment 2: investing $x_2 > 0$ dollars yields $c_2(x_2) = 3x_2 + 7$ dollars, where $c_2(0) = 0$
Investment 3: investing $x_3 > 0$ dollars yields $c_3(x_3) = 4x_3 + 5$ dollars, where $c_3(0) = 0$

(Note that if no money is invested, then there is no yield; the yield is non-linear.)

Suppose that we can invest \$6,000 and each investment must be a multiple of \$1,000. We identify stages

stages: in stage i, we consider only investments $1, \ldots, i$.

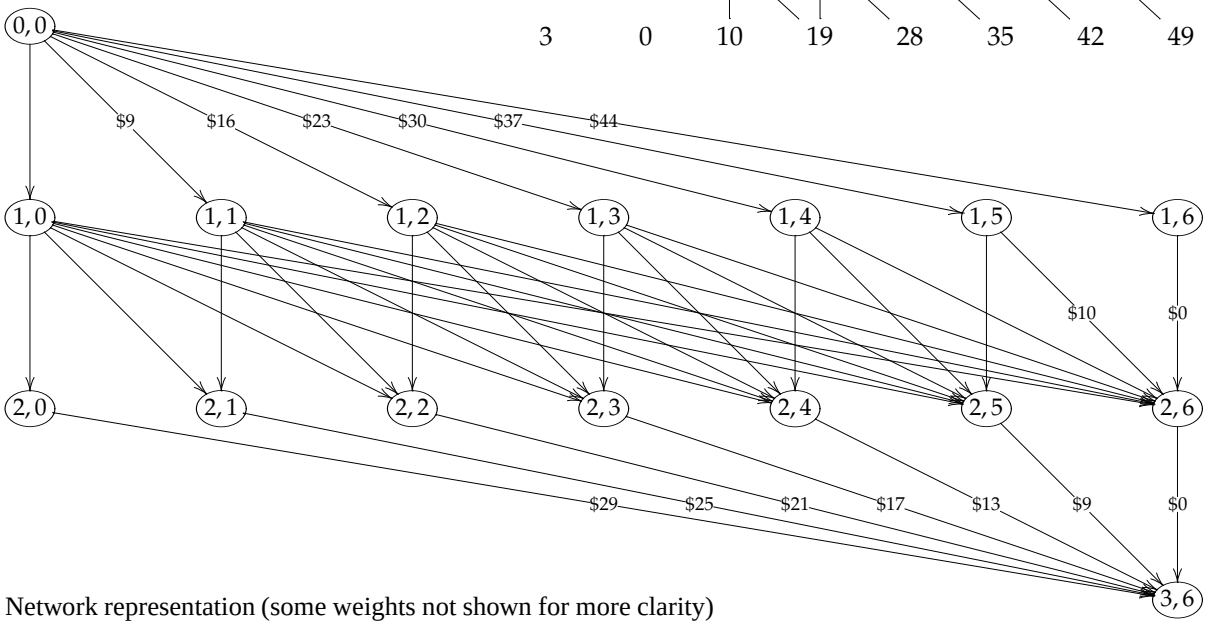
states: the available budget as a multiple of \$1,000, up to \$6,000.

values: $f_i(j) =$ maximum yield we can get by investing j thousands of \$ into investments #1,...,#i.

recursion: $f_i(j) = \begin{cases} 0 & i = 0 \\ \max\limits_{k \in \{0,1,\ldots,j\}} \left\{ c_i(k) + f_{i-1}(j-k) \right\} & i \geq 1 \end{cases}$

Note: the arrow indicates from which subproblem was an optimal solution obtained

stage	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	9	16	23	30	37	44
2	0	10	19	26	33	40	47
3	0	10	19	28	35	42	49



Network representation (some weights not shown for more clarity)

Inventory problem

A company must meet demand in the next four months as follows: month 1: $d_1 = 1$ unit, month 2: $d_2 = 3$ units, month 3: $d_3 = 2$ units, month 4: $d_4 = 4$ units. At the beginning of each month it has to be determined how many units to produce. Production has a setup cost of \$3 and then \$1 for each unit. At the end of the month there is holding cost \$0.50 for each unit at hand. The company's warehouse can store up to 4 units from month to month. The capacity

of the production line allows at most 5 units to be produced each month. Initially no products at hand. Determine the minimum cost of production that meets the demand.

stages: production until (and including) month i

states: number of units at hand at the end of month i

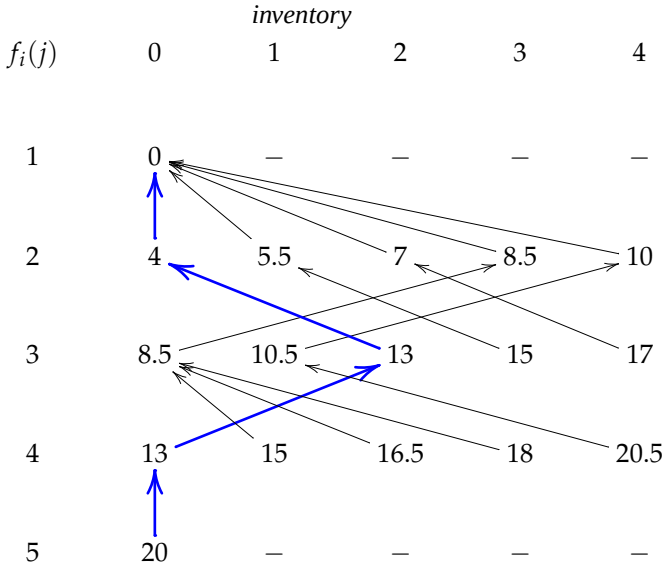
values: $f_i(j) =$ the minimum cost of production that ends in month i with j units in the inventory

For example, if we have 4 units at the end of month 2, then we meet month 3 demand of 3 units if we either

- do not produce and are left with 1 unit, held for \$0.50 **total cost \$0.50**
- pay setup cost \$3, produce 1 units for \$1, and are left with 2 units, held for \$1 **total cost \$5.00**
- pay setup cost \$3, produce 2 units for \$2, and are left with 3 units, held for \$1.50 **total cost \$6.50**
- pay setup cost \$3, produce 3 units for \$3, and are left with 4 units, held for \$2 **total cost \$8.00**

Note that in our calculation with already include the holding cost.

$$f_i(j) = \underbrace{j \times \$0.50}_{\text{holding cost}} + \min \begin{cases} f_{i-1}(j+d_i) & \text{no production } k = 0 \\ \underbrace{\$3 + k \times \$1}_{\text{production cost}} + f_{i-1}(j+d_i-k) & \text{production } k \in \{1,2,\ldots,j+d_i\} \end{cases}$$



Shortest paths

All-pairs Shortest Paths: (Floyd-Warshall) given a network $G = (V, E)$ where each edge $(u, v) \in E$ has length/cost c_{uv}, determine the distance between every pair of nodes.

We can solve this problem using dynamic programming as follows. We label the vertices $v_1, v_2, \ldots, v_n$.

stages: in stage i consider only shortest paths going through intermediate nodes $v_1, \ldots, v_i$

values: $f_i(u, v) =$ minimum length of path from u to v whose all intermediate nodes are among $v_1, \ldots, v_i$.

$$f_0(u,w) = \begin{cases} 0 & u = w \\ c_{uw} & uw \in E \\ \infty & uw \notin E \end{cases}$$

$$f_i(u,w) = \min\left\{ f_{i-1}(u,w),\ f_{i-1}(u,v_i) + f_{i-1}(v_i,w) \right\} \quad \text{for } i \geq 1$$

Single-source Shortest Paths: (Bellman-Ford) find distances from a fixed source s to all other vertices

stages: in stage i consider only shortest paths using at most i edges

values: $f_i(u) =$ minimum length of a path going from s to u and having at most i edges

We have $f_0(s) = 0$ and $f_0(u) = \infty$ for all $u \neq s$, since we cannot use any edge at this stage. For later stages:

$$f_i(u) = \min\left\{f_{i-1}(u),\ \min_{\substack{v\in V\\ vu\in E}}\left\{f_{i-1}(v) + c_{vu}\right\}\right\}$$

Knapsack revisited

$$\begin{aligned} \max\ & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ & d_1x_1 + d_2x_2 + \ldots + d_nx_n \leq B \\ & x_1, \ldots, x_n \in \{0,1\} \end{aligned}$$

$f_i(j)$ = optimal solution using first i items and bag of size j

$$f_i(j) = \begin{cases} -\infty & j < 0 \\ 0 & i = 0 \\ \max\left\{f_{i-1}(j),\ c_i + f_{i-1}(j-d_i)\right\} & i \geq 1 \end{cases}$$

Alternative recursion (only for unbounded Knapsack)

Item	Weight (lbs)	Price ($)
1	4	11
2	3	7
3	5	12

$\rightarrow$ fill a 10 pound bag

$$\begin{aligned} \max\ & 11x_1 + 7x_2 + 12x_3 \\ & 4x_1 + 3x_2 + 5x_3 \leq 10 \\ & x_1, x_2, x_3 \geq 0 \text{ and integer} \end{aligned}$$

$g(j)$ = the maximum total price obtained by filling a j pound bag

recursion formula: $g(j) = \max\limits_{\substack{i\in\{1,2,3\}\\ j\geq d_i}}\left\{c_i + g(j-d_i)\right\}$

We try to put item i into the bag and fill the rest as best as possible (note that here we need that we have unlimited number of each item rather than just one; it would not work for binary Knapsack).

j	0	1	2	3	4	5	6	7	8	9	10
$g(j)$	0	0	0	7	11	12	14	18	22	23	25

$g(0) = 0$
$g(1) = 0$
$g(2) = 0$
$g(3) = \max\{c_2 + g(3-d_2)\} = c_2 + g(0) = 7 + 0 = 7$

$$g(4) = \max\begin{cases} c_1 + g(0) = 11 + 0 = 11^* \\ c_2 + g(1) = 7 + 0 = 7 \end{cases}$$

$$g(5) = \max\begin{cases} c_1 + g(1) = 11 + 0 = 11 \\ c_2 + g(2) = 7 + 0 = 7 \\ c_3 + g(0) = 12 + 0 = 12^* \end{cases}$$

$$g(6) = \max\begin{cases} c_1 + g(2) = 11 + 0 = 11 \\ c_2 + g(3) = 7 + 7 = 14^* \\ c_3 + g(1) = 12 + 0 = 12 \end{cases}$$

$$g(7) = \max\begin{cases} c_1 + g(3) = 11 + 7 = 18^* \\ c_2 + g(4) = 7 + 11 = 18^* \\ c_3 + g(2) = 12 + 0 = 12 \end{cases}$$

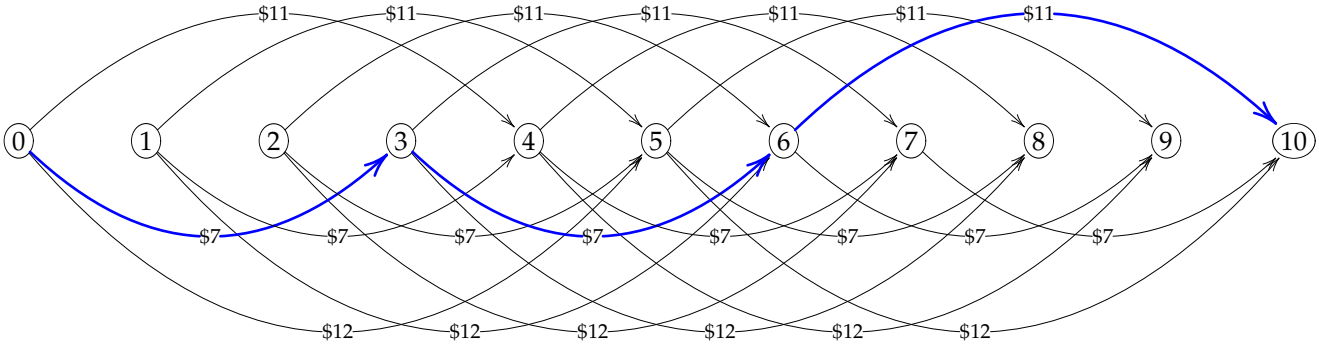
$$g(8) = \max\begin{cases} c_1 + g(4) = 11 + 11 = 22^* \\ c_2 + g(5) = 7 + 12 = 19 \\ c_3 + g(3) = 12 + 7 = 19 \end{cases}$$

$$g(9) = \max\begin{cases} c_1 + g(5) = 11 + 12 = 23^* \\ c_2 + g(6) = 7 + 14 = 21 \\ c_3 + g(4) = 12 + 11 = 23^* \end{cases}$$

$$g(10) = \max\begin{cases} c_1 + g(6) = 11 + 14 = 25^* \\ c_2 + g(7) = 7 + 18 = 25^* \\ c_3 + g(5) = 12 + 12 = 24 \end{cases}$$

Optimal solution: $g(10) = c_1 + g(6) = c_1 + (c_2 + g(3)) = c_1 + c_2 + (c_2 + g(0)) = c_1 + c_2 + c_2$

Best solution for a 10 pound bag: pick 2 items #2 and 1 item #1 for a total value of \$25.



Optimal solution (longest path) shown in **blue** color.

Scheduling

We have 7 jobs $j_1, j_2, \ldots, j_7$ with processing times $p_1, p_2, \ldots, p_7$ given as follows:

job j_i	j_1	j_2	j_3	j_4	j_5	j_6	j_7
processing time p_i	10	8	6	5	4	4	3

We wish to schedule all 7 jobs on 2 machines. The goal is to minimize the completion time of all jobs.

n jobs $j_1, \ldots, j_n$
processing times $p_1, \ldots, p_n$

m machines

$$x_{ij} = \begin{cases} 1 & \text{if job } i \text{ scheduled on machine } j \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} \min \quad & z \\ & \sum_j x_{ij} = 1 \qquad i = 1, \ldots, n \\ & \sum_i p_i x_{ij} \le z \qquad j = 1, \ldots, m \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

stages: at stage i we schedule first i jobs $j_1, \ldots, j_i$

state: pair (t_1, t_2) denoting used up time t_1 on machine #1 and time t_2 on machine #2

value: $f_i(t_1, t_2) = 1$ if it is possible to schedule first i jobs on the two machines using t_1 time on machine #1 and t_2 time on machine #2;

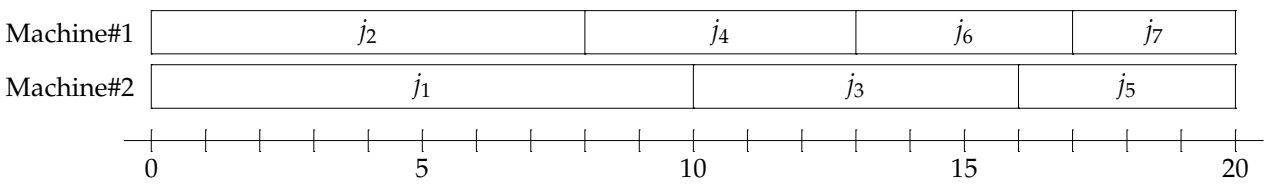
$f_i(t_1, t_2) = 0$ if otherwise

(Note that the order of the jobs does not matter here; only on which machine each job is executed.)

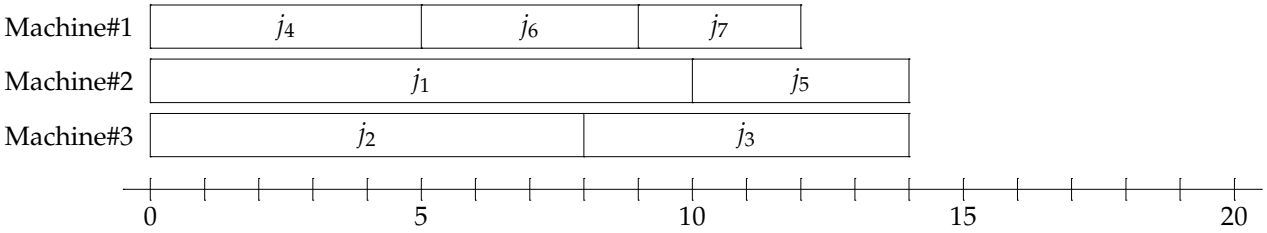
recursion:
$$f_i(t_1, t_2) = \begin{cases} 0 & t_1 < 0 \text{ or } t_2 < 0 \\ 1 & i = 0 \\ \max\left\{ f_{i-1}(t_1 - p_i, t_2),\ f_{i-1}(t_1, t_2 - p_i) \right\} & i \ge 1 \end{cases}$$

We either execute job j_i on machine #1 or on machine #2. We pick better of the two options.

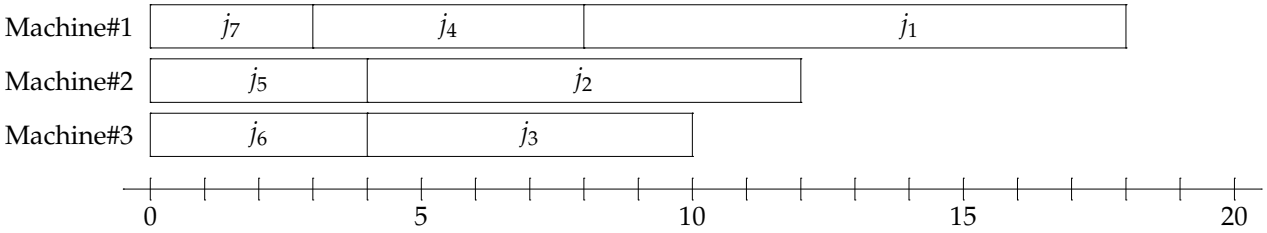
answer: is given by taking smallest t such that $f_7(t, t) = 1$



(How would you formulate the problem for 3 machines instead of 2?)



Average Completion time: longest job last, assign to machines in a round robin fashion → optimal



Average completion time = $8\frac{3}{7} \sim 8.43$ (note that moving j_7 to any other machine does not change the average)

Traveling Salesman

A small company delivers from New York to its customers in neighboring cities (Washington, Pittsburgh, Buffalo, and Boston). The distances (in miles) between cities are as follows.

New York (NY)	228	372	396	211
	Washington (WA)	244	381	437
		Pittsburgh (PI)	219	571
			Buffalo (BU)	452
				Boston (BO)

The company owns one delivery truck. In order to deliver to customers in all the 4 cities, the truck departs from New York and visits each city in turn before heading back to New York. To save on fuel, the company wants to find a route that minimizes the total distance the truck has to travel.

We can think of the route as follows. At each point of our journey, we are in some city v and before coming to v we have visited (traveled through) all cities in S (a subset of cities). Note that neither city v nor New York (our starting city) is in the set S.

stage: in stage S, we have travelled through all cities in S (a selected subset of cities)

state: city v, our current position on the route (after visiting all cities in S)

value: $f(S,v) =$ the minimum cost of travelling from New York to v while visiting all cities in S

Our goal is to find $f\Big(\{WA, PI, BU, BO\}, NY\Big)$. That is, we visit all cities and come back to New York.

In order to do so, we compute $f(S,v)$ for all possible values of S and v. **How do we do that?** Observe that if we visited all cities in S and then arrived to v, we must have arrived to v from some city u in the set S. The total distance of such a journey is then the distance from u to v plus the minimum distance we need to travel in order to arrive to u while visiting all cities in $S \setminus \{u\}$. We try all possible choices for u and select the one that minimizes the total distance.

Let $c(u,v)$ denote the distance from city u to city v. Then the recursion is as follows:

recursion: $f(S,v) = \min_{u \in S} \Big(f(S \setminus \{u\}, u) + c(u,v)\Big)$

initial conditions: $f(\emptyset, v) = c(NY, v)$

We calculate the values of $f(S,v)$ for smaller sets first and then gradually for bigger sets. Having done so in this order, we can **reuse** the values we calculated earlier (for smaller sets). This is the **main principle** of dynamic programming. Let us calculate the values.

$S \setminus v$	WA	PI	BU	BO	NY
$\emptyset$	228	372	396	211	
$\{WA\}$		472	609	665	
$\{PI\}$	616		519	943	
$\{BU\}$	777	615		848	
$\{BO\}$	648	782	663		
$\{WA,PI\}$			691	1043	
$\{WA,BU\}$		828		1061	
$\{WA,BO\}$	892	1029			
$\{PI,BU\}$	859			1043	
$\{PI,BO\}$	1026		1001		
$\{BU,BO\}$	1115	882			
$\{WA,PI,BU\}$				1143	
$\{WA,PI,BO\}$			1111		
$\{WA,BU,BO\}$		1248			
$\{PI,BU,BO\}$	1126				
$\{WA,PI,BU,BO\}$					1354

$f(\emptyset,WA) = c(NY,WA) = 228$
$f(\emptyset,PI) = c(NY,PI) = 372$
$f(\emptyset,BU) = c(NY,BU) = 396$
$f(\emptyset,BO) = c(NY,BO) = 211$
$f(\{WA\},PI) = \min\{f(\emptyset,WA) + c(WA,PI)\}$
$= 228 + 244 = 472$
$f(\{WA\},BU) = \min\{f(\emptyset,WA) + c(WA,BU)\}$
$= 228 + 381 = 609$
$f(\{WA\},BO) = \min\{f(\emptyset,WA) + c(WA,BO)\}$
$= 228 + 437 = 665$
$f(\{PI\},WA) = \min\{f(\emptyset,PI) + c(PI,WA)\}$
$= 372 + 244 = 616$
$f(\{PI\},BU) = \min\{f(\emptyset,PI) + c(PI,BU)\}$
$= 372 + 219 = 591$
$f(\{PI\},BO) = \min\{f(\emptyset,PI) + c(PI,BO)\}$
$= 372 + 571 = 943$

$f(\{BU\},WA) = \min\{f(\emptyset,BU) + c(BU,WA)\} = 396 + 381 = 777$
$f(\{BU\},PI) = \min\{f(\emptyset,BU) + c(BU,PI)\} = 396 + 219 = 615$
$f(\{BU\},BO) = \min\{f(\emptyset,BU) + c(BU,BO)\} = 396 + 452 = 848$
$f(\{BO\},WA) = \min\{f(\emptyset,BO) + c(BO,WA)\} = 211 + 437 = 648$
$f(\{BO\},PI) = \min\{f(\emptyset,BO) + c(BO,PI)\} = 211 + 571 = 782$
$f(\{BO\},BU) = \min\{f(\emptyset,BO) + c(BO,BU)\} = 211 + 452 = 663$

$$f(\{WA,PI\},BU) = \min\begin{cases} f(\{PI\},WA) + c(WA,BU) = 616 + 381 = 997 \\ f(\{WA\},PI) + c(PI,BU) = 472 + 219 = 691^* \end{cases}$$

$$f(\{WA,PI\},BO) = \min\begin{cases} f(\{PI\},WA) + c(WA,BO) = 616 + 437 = 1053 \\ f(\{WA\},PI) + c(PI,BO) = 472 + 571 = 1043^* \end{cases}$$

$$f(\{WA,BU\},PI) = \min\begin{cases} f(\{BU\},WA) + c(WA,PI) = 777 + 244 = 1021 \\ f(\{WA\},BU) + c(BU,PI) = 609 + 219 = 828^* \end{cases}$$

$$f(\{WA,BU\},BO) = \min\begin{cases} f(\{BU\},WA) + c(WA,BO) = 777 + 437 = 1214 \\ f(\{WA\},BU) + c(BU,BO) = 609 + 452 = 1061^* \end{cases}$$

$$f(\{WA,BO\},PI) = \min\begin{cases} f(\{BO\},WA) + c(WA,PI) = 648 + 244 = 892^* \\ f(\{WA\},BO) + c(BO,PI) = 665 + 571 = 1236 \end{cases}$$

$$f(\{WA,BO\},BU) = \min\begin{cases} f(\{BO\},WA) + c(WA,BU) = 648 + 381 = 1029^* \\ f(\{WA\},BO) + c(BO,BU) = 665 + 452 = 1117 \end{cases}$$

$$f(\{PI,BU\},WA) = \min\begin{cases} f(\{PI\},BU) + c(BU,WA) = 591 + 381 = 972 \\ f(\{BU\},PI) + c(PI,WA) = 615 + 244 = 859^* \end{cases}$$

$$f(\{PI,BU\},BO) = \min\begin{cases} f(\{PI\},BU) + c(BU,BO) = 591 + 452 = 1043^* \\ f(\{BU\},PI) + c(PI,BO) = 615 + 571 = 1186 \end{cases}$$

$$f(\{PI, BO\}, WA) = \min \begin{cases} f(\{PI\}, BO) + c(BO, WA) = 943 + 437 = 1380 \\ f(\{BO\}, PI) + c(PI, WA) = 782 + 244 = 1026^* \end{cases}$$

$$f(\{PI, BO\}, BU) = \min \begin{cases} f(\{PI\}, BO) + c(BO, BU) = 943 + 452 = 1395 \\ f(\{BO\}, PI) + c(PI, BU) = 782 + 219 = 1001^* \end{cases}$$

$$f(\{BU, BO\}, WA) = \min \begin{cases} f(\{BU\}, BO) + c(BO, WA) = 848 + 437 = 1285 \\ f(\{BO\}, BU) + c(BU, WA) = 663 + 452 = 1115^* \end{cases}$$

$$f(\{BU, BO\}, PI) = \min \begin{cases} f(\{BU\}, BO) + c(BO, PI) = 848 + 571 = 1419 \\ f(\{BO\}, BU) + c(BU, PI) = 663 + 219 = 882^* \end{cases}$$

$$f(\{WA, PI, BU\}, BO) = \min \begin{cases} f(\{WA, PI\}, BU) + c(BU, BO) = 691 + 452 = 1143^* \\ f(\{WA, BU\}, PI) + c(PI, BO) = 828 + 571 = 1399 \\ f(\{PI, BU\}, WA) + c(WA, BO) = 859 + 437 = 1296 \end{cases}$$

$$f(\{WA, PI, BO\}, BU) = \min \begin{cases} f(\{WA, PI\}, BO) + c(BO, BU) = 1043 + 452 = 1495 \\ f(\{WA, BO\}, PI) + c(PI, BU) = 892 + 219 = 1111^* \\ f(\{PI, BO\}, WA) + c(WA, BU) = 1026 + 381 = 1407 \end{cases}$$

$$f(\{WA, BU, BO\}, PI) = \min \begin{cases} f(\{WA, BU\}, BO) + c(BO, PI) = 1061 + 571 = 1632 \\ f(\{WA, BO\}, BU) + c(BU, PI) = 1029 + 219 = 1248^* \\ f(\{BU, BO\}, WA) + c(WA, PI) = 1115 + 244 = 1359 \end{cases}$$

$$f(\{PI, BU, BO\}, WA) = \min \begin{cases} f(\{PI, BU\}, BO) + c(BO, WA) = 1043 + 437 = 1480 \\ f(\{PI, BO\}, BU) + c(BU, WA) = 1001 + 381 = 1382 \\ f(\{BU, BO\}, PI) + c(PI, WA) = 882 + 244 = 1126^* \end{cases}$$

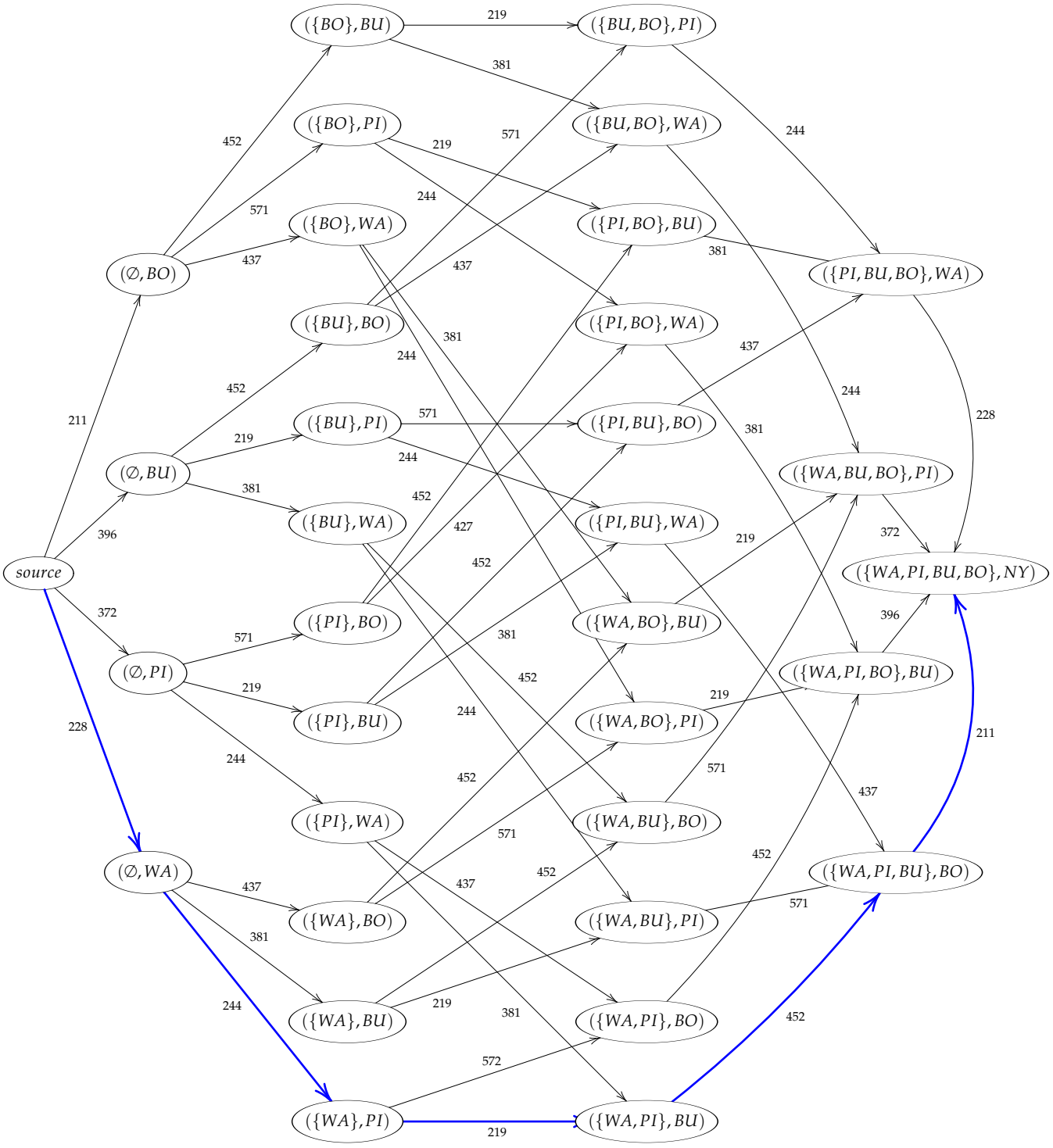
$$f(\{WA, PI, BU, BO\}, NY) = \min \begin{cases} f(\{WA, PI, BU\}, BO) + c(BO, NY) = 1143 + 211 = 1354^* \\ f(\{WA, PI, BO\}, BU) + c(BU, NY) = 1111 + 396 = 1507 \\ f(\{WA, BU, BO\}, PI) + c(PI, NY) = 1248 + 372 = 1620 \\ f(\{PI, BU, BO\}, WA) + c(WA, NY) = 1126 + 228 = 1354^* \end{cases}$$

Optimal solution:

$$\begin{aligned} f(\{WA, PI, BU, BO\}, NY) &= f(\{WA, PI, BU\}, BO) + c(BO, NY) \\ &= f(\{WA, PI\}, BU) + c(BU, BO) + c(BO, NY) \\ &= f(\{WA\}, PI) + c(PI, BU) + c(BU, BO) + c(BO, NY) \\ &= f(\emptyset, WA) + c(WA, PI) + c(PI, BU) + c(BU, BO) + c(BO, NY) \\ &= c(NY, WA) + c(WA, PI) + c(PI, BU) + c(BU, BO) + c(BO, NY) \end{aligned}$$

Travel from New York to Washington to Pittsburgh to Buffalo to Boston and back.

The **total travel distance** 1354 miles.



The cost on edges indicates the increase in distance when that particular action (decision) is taken.

Optimal solution shown in **blue** – longest path from *source* to $(\{WA, PI, BU, BO\}, NY)$.

14 Analysis of efficiency

In the previous chapters, we have discussed various computational methods for optimization problems such as finding optimal solutions to linear programs (Simplex), finding optimal paths and flows in networks (Dijkstra, Ford-Fulkerson, Network Simplex), solving integer linear programs (Cutting planes, Branch-and-Bound), and finally optimization using recursive methods leveraging memory (dynamic programming).

Some of these methods allowed us to solve the same problem using different approaches (for instance, we can solve shortest paths using the Simplex method, or Dijkstra's algorithm, or dynamic programming). In order to understand the benefits of these different methods, we need to be able to compare them in a uniform way. (We will be deliberately vague; a rigorous treatment of this subject is beyond the scope of this text.)

Let us first discuss in rough numbers the number of steps (operations) of each of the methods.

Dynamic Programming

- **stages** $1, 2, \ldots, T$
- **states** $1, 2, \ldots, M$ (for each stage)
- **next** stage computed from values of **previous** stage
- **recursive formula** for the value of each state
- **objective** is the value of a specific state in the **last** stage T
- using the recursive formula, we calculate the value for **every stage** and **every state**
  - M stages
  - T states
  - altogether $M \times T$ states to evaluate
  - for each state and stage we look at all M states of the previous stage and calculate the best answer $\rightarrow$ M calculations

  altogether $M^2 \times T$ calculations using M^2 memory

Simplex algorithm

The number of steps is proportional to the number of **bases** (dictionaries) we go through. To do so, we need to make sure not to encounter the same basis twice during the calculation (for instance, by Bland's rule).

- n variables
- m equations ($n \geq m$)
- each basis consists of a selection of m variables $\rightarrow$ at most $\binom{n}{m}$ different bases

Therefore Simplex method takes at most $\binom{n}{m}$ pivotting steps. This is roughly n^m for small m, but around 2^n for large m (say $m = n/2$). Note that this seems like a very loose (and pessimistic) estimate. Does the worst-case really happen? Can we get a better general estimate? Unfortunately, there are examples which exhibit this worst-case behaviour $\rightarrow$

Klee-Minty examples:

$$\max \quad \sum_{j=1}^{n} 10^{n-j} x_j$$

$$\text{s.t.} \quad \left(2 \sum_{j=1}^{i-1} 10^{i-j} x_j\right) + x_i \leq 100^{i-1} \qquad (i = 1, 2, \ldots, n)$$

$$x_j \geq 0 \qquad (j = 1, 2, \ldots, n)$$

for $n = 3$, this looks as follows

$$\begin{aligned} \max \quad & 100x_1 + 10x_2 + x_3 \\ \text{s.t.} \quad & x_1 \leq 1 \\ & 20x_1 + x_2 \leq 100 \\ & 200x_1 + 20x_2 + x_3 \leq 10{,}000 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

If at every step the entering variable is chosen to be the one with **largest coefficient** in z, then the Klee-Minty examples go through $2^n - 1$ bases before finding the optimum.

Branch-and-Bound

- each subproblem is a linear program (solved by Simplex or other methods)
- only bounds on variables change $\rightarrow$ **size** of the LP is the **same** in each subproblem
- possibly 2^n subproblems (unavoidable in general, even if we branch exactly once on each variable)

Cutting Planes

- at each point we have **only one** linear program (no subproblems)
- each step adds one new constraint and one new variable
- the linear program **grows** at each step
- possibly 2^n steps before optimum reached (unavoidable in general)
- could be much worse than Branch-and-Bound once the size of the LP becomes too big (recall that with BnB the LP remains the same size in all subproblems)

Network Algorithms

Dijkstra

- n nodes, m edges $\rightarrow$ n steps (one for each node)
- each step involves: finding a smallest value $d(u)$, and updating other values $d(v)$ $\rightarrow$ roughly $\approx 2n$ calculations

altogether $\approx 2n^2$ operations (can be improved to $\approx m \log n$ with special data structures)

Ford-Fulkerson (Augmenting path algorithm)

- n nodes, m edges
- each step **constructs** the residual network, finds an **augmenting path** and augments the flow $\rightarrow$ roughly $\approx 2(n+m)$ operations for each step
- at most $n \times m$ steps needed if shortest augmenting path is used (Edmonds-Karp)

altogether $\approx 2n^2m$ operations needed (can be improved to $\approx nm$ by additional tricks)

Network Simplex (Cycle cancelling algorithm)

- n nodes, m edges
- each step **calculates** shadow prices and reduced costs, finds a cycle (**loop**) and adjusts the flow $\rightarrow$ roughly $\approx 2n+m$ operations for each step
- at most $\approx nm \log(Cn)$ steps needed if minimum mean-cost cycle is used (Goldberg-Tarjan) where C is largest cost of an edge

altogether $\approx 2n^2m \log(Cn)$ operations needed (can be improved to $\approx nm$ by additional tricks)

Other Network Problems

n nodes, m edges

1. **Transportation Problem:** same as Network Simplex $\approx nm$
2. **Assignment Problem:** the number of required steps can be shown to be at most $\sqrt{n}$
   $\rightarrow$ altogether $\approx \sqrt{n}m$ operations need
3. **All-pairs shortest paths (Floyd-Warshall):**
   - nodes are labeled $v_1, \ldots, v_n$
   - at each step i we improve estimate $d(u,v)$ on the distance between u and v by considering paths from u to v passing through v_i (and some of the $v_1, \ldots, v_{i-1}$) $\rightarrow n^3$ calculations in each step

   altogether n^4 operations
4. **Single-source shortest path (Bellman-Ford)** with general edge costs (negative costs allowed)
   - at step i we consider paths using i edges $\rightarrow m$ calculations in each step

   altogether $n \times m$ operations

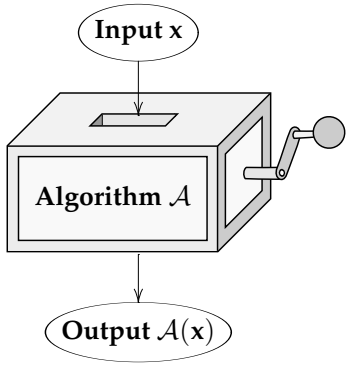
14.1 Algorithmic Complexity

a way to compare efficiency (speed) of different computational methods (algorithms) and corresponding computational problems. **How to measure efficiency of algorithms?** Standard practice is to ask:

"**How does** the **algorithm scale** when **inputs** get **large**?"

i.e., how much **time** or **memory** $f(n)$ it takes to compute a solution if the input data has size n.

Algorithm



An *algorithm* $\mathcal{A}$
- takes an **input x** (numbers),
- performs a certain number of operations (additions, multiplications),
- produces an **output** $\mathcal{A}(\mathbf{x})$ (a number, answer yes/no)

time complexity = number of (elementary) operations performed

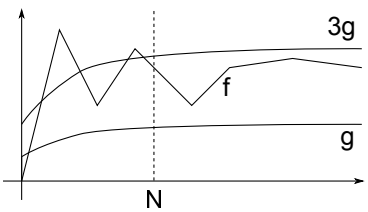
space complexity = size of memory used during the computation

time $\geq$ **space**

$f(n) =$ **worst case** time complexity of $\mathcal{A}$ for inputs of size n
(maximum number of operations that $\mathcal{A}$ performs on any input of size n)

Asymptotic notation

We are interested in the behaviour of $f(n)$ as n goes to infinity. We use the following asymptotic notation.



For two functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$, we say that f is $O(g)$ "the order of g", and write $f = O(g)$, if there exist constants $c > 0$ and $N > 0$ such that

$$f(n) \leq c \cdot g(n) \text{ for all } n > N.$$

Example: let $f(n) = n^2 + n + 1$ and $g(n) = n^2$. Then $f = O(g)$, since for $c = 2$ and $n > N = 2$ we have

$$f(n) = n^2 + n + 1 \leq 2 \cdot g(n) = 2n^2$$

Growth of functions

We need to highlight the growth of various complexity functions we shall encounter.

$f(n)\backslash n$	2	3	5	10	20	50	100	1 000	10 000	100 000	1 000 000	10 000 000
$\log_2 n$	1	2	2	3	4	6	7	10	13	17	20	23
n	2	3	5	10	20	50	100	1 000	10 000	100 000	1 000 000	10 000 000
$n \log_2 n$	2	5	12	33	86	282	664	9 966	132 877	1 660 964	19 931 569	232 534 967
$n^{1.5}$	3	5	11	32	89	354	1 000	31 623	1 000 000	31 622 777	1 000 000 000	31 622 776 602
n^2	4	9	25	100	400	2 500	10 000	1 000 000	100 000 000	10 000 000 000	1 000 000 000 000	100 000 000 000 000
$n^{2.5}$	6	16	56	316	1 789	17 678	100 000	31 622 777	10 000 000 000	3 162 277 660 168	1 000 000 000 000 000	316 227 766 016 837 933
n^3	8	27	125	1 000	8 000	125 000	1 000 000	1 000 000 000	1 000 000 000 000	1 000 000 000 000 000	1 000 000 000 000 000 000	
n^4	16	81	625	10 000	160 000	6 250 000	100 000 000	1 000 000 000 000	10 000 000 000 000 000	100 000 000 000 000 000 000		
2^n	4	8	32	1 024	1 048 576	1 125 899 906 842 624	1 267 650 600 228 229 401 496 703 205 376					
$n!$	2	6	120	3 628 800	2 432 902 008 176 640 000	30 414 093 201 713 378 043 612 608 166 064 768 844 377 641 568 960 512 000 000 000 000						
n^n	4	27	3 125	10 000 000 000	104 857 600 000 000 000 000 000 000	8 881 784 197 001 252 323 389 053 344 726 562 500 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000						

1. The first block are what is considered **fast** or **practical** algorithms (can handle inputs of size billions).
2. The second block are **efficient** algorithms (inputs can range up to size millions).
3. The last block are algorithms for **hard** problems (can only handle inputs up to size hundred or so).

Summary of complexity of selected problems

In the following, the input to each of the problems will consist of

- n numbers (or $n + m$ numbers, or $n \times m$ numbers where $n \geq m$ in case of LP)
- L will denote the number of bits needed to represent any of these numbers
  (in a typical computer using IEEE754 floating point numbers $L = 24$, 53, 64, or 113)

Problem	Time complexity	Space complexity
Linear Programming (Simplex): n variables, m constraints	$2^{O(n)}$	$O(n \cdot m)$
Linear Programming (Interior point): n variables, m constraints	$O(n^3 L)$	$O(n \cdot m)$
Shortest path (Dijkstra): n nodes, m edges	$O(m + n \cdot \log(n))$	$O(n)$
Shortest path (Bellman-Ford): n nodes, m edges	$O(m \cdot n)$	$O(n)$
All-pairs Shortest paths (Floyd-Warshall): n nodes, m edges	$O(n^4)$	$O(n^2)$
Minimum Spanning tree (Prim, Kruskal): n nodes, m edges	$O(m \cdot \log(n))$	$O(m)$
Assignment problem: n nodes, m edges	$O(\sqrt{n}m) = O(n^{2.5})$	
Maximum flow: n nodes, m edges	$O(m \cdot n) = O(n^3)$	$O(m)$
Minimum cost flow (Network Simplex): n nodes, m edges	$O(m^2 \cdot \log^2(n))$	$O(m)$
0-1 Knapsack (Integer LP): n items	$2^{O(n)}$	$O(n)$
0-1 Knapsack (Dynamic program): n items, budget B	$O(n \cdot B)$	$O(n \cdot B)$
Knapsack (Dynamic program): n items, budget B	$O(n \cdot B)$	$O(B)$
Machine Scheduling (Dynamic programming): n jobs, m machines, completion time T	$O(n \cdot m \cdot T^m)$	$O(n \cdot T^m)$
Machine Scheduling (Integer LP)	$O(m^n)$	$O(n)$

Inventory problem (Dynamic program): n months, warehouse size M	$O(n \cdot M^2)$	$O(n \cdot M)$
Inventory problem (Integer LP)	$O(M^n)$	$O(n)$

Important note: recall that when we solving linear programs we have no guarantee that the solution we obtain will be an integer point, even if there exists an optimal integer point. Finding an integer solution to a linear program is in general difficult.

However, this is not the case for Network problems (Shortest paths, Maximum flow, Minimum-cost flow).

Theorem 6. *If all capacities/source supplies/destination demands/node net supplies are integers, then*

(i) Every **basic feasible solution** *to the Linear Program for Shortest path Problem (Maximum flow Problem, Transportation Problem, Minimum-cost flow Problem) is integral (all variables are integers).*

(ii) There exists an **optimal integer** *solution (to the above problems) because one such a solution is basic.*

(iii) Moreover, Dijkstra's (Ford-Fulkerson, Transportation Simplex, Network Simplex) algorithm finds this solution efficiently (in polynomial time).

So if your problem is a Network problem (or you can turn your LP into an equivalent Network problem), then optimal integer solution can be found efficiently (unlike using purely ILP techniques).