5CCS2FC2: Foundations of Computing II

Linear Programming

Week 8

Dr Christopher Hampson

Department of Informatics

King's College London

Linear Programming

Linear Progamming

- **Linear Programming**
  - **Objective Function** The quantity to be maximised / minimised,
  - **Linear Constraints** Set of linear inequalities restricting the possible solutions.

$$
\begin{array}{rrcl}
\text{Maximise:} & 2x + 3y & & \\
\text{Subject to:} & 3x + 2y & \leq & 15 \\
 & 2y - x & \leq & 5 \\
 & x + 2y & \leq & 7 \\
 & x, y & \geq & 0
\end{array}
$$

(there may be a unique solution, infinitely many solutions or no solution)

- **Example:** A company creates its smoothies using:

Banana 🍌, Strawberry 🍓 and Mango 🥭

Each smoothie must contain at least 15% RDA of Vitamin A,

at least 80% RDA of Vitamin C, and at least 300mg of potassium.

- 100g of **Banana** contains 1% of the RDA of Vitamin A, 14% of the RDA of Vitamin C, 358mg of potassium and costs 15p.
- 100g of **Strawberry**contains no Vitamin A, 97% of the RDA of Vitamin C, 153mg of potassium and costs 27p.
- 100g of **Mango** contains 21% of the RDA of Vitamin A, 60% of the RDA of Vitamin C, 168mg of potassium and costs 30p.

What proportion of ingredients would maximise profits?

- **Example (cont.)**

Cost of Production

Vitamin A requirement

$$
\begin{array}{lrcl}
\text{Minimise:} & 15x + 27y + 30z & & \\
\text{Subject to:} & x + 21z & \geq & 15 \\
 & 14x + 97y + 60z & \geq & 80 \\
 & 358x + 153y + 168z & \geq & 300 \\
 & x,\ y,\ z & \geq & 0
\end{array}
$$

Vitamin C requirement

Potassium requirement

Non-negativity constraint

- **Example (cont.)**

Optimal Solution

$$x = 0.365, \quad y = 0.341, \quad z = 0.697$$

$\rightsquigarrow$ **Cost = 35.59p**

- In order to solve LPs, it is helpful to write them in a **Standard Form**:
  - This reduces the number of cases we must consider,
  - We can design algorithms that are highly specialised for a particular input format

(this is why we use CNF as the 'standard form' for SAT problems)

LP Standard Form

- **Standard Form for Linear Programs**
  - The criteria must be to **maximise** the objective function,
  - All **linear constrainst** must be of the form *'less-than-or-equal-to'*

$$\boxed{a_1x_1 + a_2x_2 + \cdots + a_nx_n \ \leq \ c}$$

(where $a_1, \ldots, a_n$ and c are constants)

  - We seek a **non-negative solution** with the additional constaint

$$\boxed{x_1, \ldots, x_n \ \geq \ 0}$$

LP Standard Form

- **Example:**

$$\begin{aligned} \text{Maximise:} \quad & 2x + y \\ \text{Subject to:} \quad & 2x + y \leq 2 \\ & x - y \leq 5 \\ & x, y \geq 0 \end{aligned}$$

standard form ✓

$$\begin{aligned} \text{Minimise:} \quad & 3y - 2x \\ \text{Subject to:} \quad & x + y = 7 \\ & 2y - x \geq -4 \\ & x, y \geq 0 \end{aligned}$$

not standard form ✗

$$\begin{aligned} \text{Maximise:} \quad & 2x - 3y \\ \text{Subject to:} \quad & x + y \leq 7 \\ & -x - y \leq -7 \\ & x - 2y \leq 4 \\ & x, y \geq 0 \end{aligned}$$

But how do we convert to Standard Form?

LP Standard Form

- **Converting to Stanard Form**

Step 1) Change the criteria for the objective function (if requried),

Minimise: $a_1x + a_2y$ $\Longrightarrow$ Maximise: $-a_1x + -a_2y$

(minimising F is the same as maximising $-F$)

Step 2) Replace any 'equality constraints'

Subject to: $b_1x + b_2y = c$ $\Longrightarrow$ Subject to: $b_1x + b_2y \leq c$
$b_1x + b_2y \geq c$

($A = c$ if and only if $A \leq c$ and $A \geq c$)

- **Converting to Stanard Form (cont.)**

  Step 3) Negate any *'greater-than-or-equal-to'* constraints

Subject to: $\boldsymbol{b}_1\boldsymbol{x} + \boldsymbol{b}_2\boldsymbol{y} \geq \boldsymbol{c}$	$\Rightarrow$	Subject to: $-\boldsymbol{b}_1\boldsymbol{x} + -\boldsymbol{b}_2\boldsymbol{y} \leq -\boldsymbol{c}$

($A \geq c$ if and only if $-A \leq -c$)

LP Standard Form

- **Converting to Stanard Form (cont.)**

Step 4) Ensure all variables are required to be *non-negative*.

(this may require introducing additional variables)

Case 4.1) If we have the constraint like $\boxed{x \leq 0}$

Replace	**Add constraint**
$x := (-x')$	$x' \geq 0$

Case 4.2) If there is no constraint on a variable x at all

Replace	**Add constraints**
$x := (x' - x'')$	$x', x'' \geq 0$

LP Standard Form

- **Slack Form**

Step 1) Replace any `*less-than-or-equal* constraints.

$$\boxed{\text{Subject to:} \quad \boldsymbol{b_1 x + b_2 y \leq c}} \quad \Longrightarrow \quad \boxed{\text{Subject to:} \quad \boldsymbol{b_1 x + b_2 y + s = c}}$$

Step 2) Replace any `*greater-than-or-equal* constraints.

$$\boxed{\text{Subject to:} \quad \boldsymbol{b_1 x + b_2 y \geq c}} \quad \Longrightarrow \quad \boxed{\text{Subject to:} \quad \boldsymbol{b_1 x + b_2 y - s = c}}$$

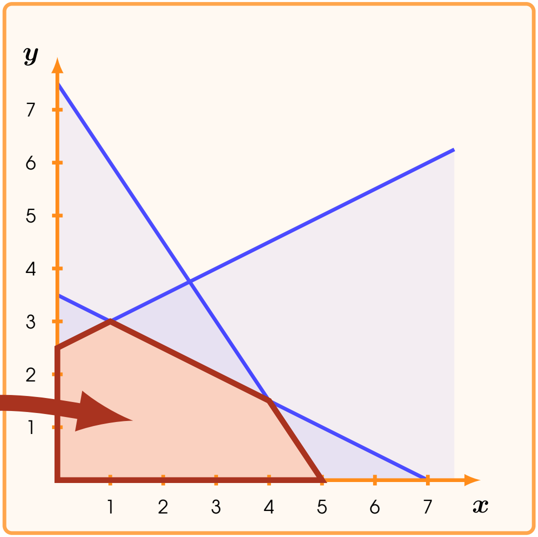
(s is an addition *slack variable*)

The Simplex Method

Visualising a Solution

- What does a solution to an LP look like?

$$
\begin{array}{lrcl}
\text{Maximise:} & 2x + 3y & & \\
\text{Subject to:} & 3x + 2y & \leq & 15 \\
 & 2y - x & \leq & 5 \\
 & x + 2y & \leq & 7 \\
 & x, y & \geq & 0
\end{array}
$$

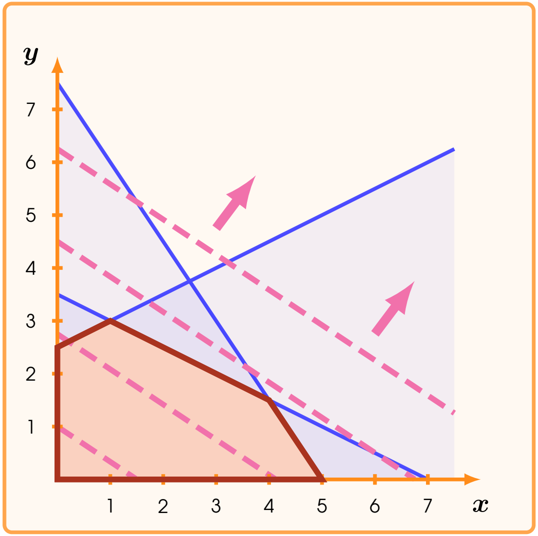


Feasible Region

Visualising a Solution

- The **Feasible Region** is the set of all possible solutions satisfying the linear constraints.

- Since the objective function is **linear**, its maximum / minimum value must occur along the **boundary** of the feasible region.



- In fact, it is enough to examine only the **corners** of the feasible region.

- If the objective function is **parallel** to some contraint, there may be **infinitely many** solutions along some edge.

The Simplex Method

- The **Simplex Method** greedily explores the boundary of the feasible region to find an optimal solution.

The Simplex Method (concept)

- Start at the origin, with all variables set to zero,
- Select the variable that leads to the greatest increase in the objective function,
- Increase until you hit a constraint,
- Move along the constraint if doing so leads to an increase in the objective function,
- Repeat, moving around the boundary of the feasible region.

Introducing Tableaux

- A **Tableau** *(plural* **Tableaux***)* is a matrix representation of a system of linear equations:

Linear Constraints

$$\begin{aligned} 3x + 2y + s_1 &= 15 \\ -x + 2y + s_2 &= 5 \\ x + 2y + s_3 &= 7 \end{aligned}$$

Objective Function

$$C = 2x + 3y$$

rewriting as
$-2x - 3y + C = 0$

matrix of coefficients

$$\begin{array}{c} \begin{array}{cccccccc} x & y & s_1 & s_2 & s_3 & C & & \end{array} \\ \left[\begin{array}{cccccc:c} 3 & 2 & 1 & 0 & 0 & 0 & 15 \\ -1 & 2 & 0 & 1 & 0 & 0 & 5 \\ 1 & 2 & 0 & 0 & 1 & 0 & 7 \\ \hdashline -2 & -3 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \end{array}$$

Introducing Tableaux

- **The Simplex Method**

Step 1) Construct the **initial tableau** from the slack form of the linear program.

$$
\begin{array}{c}
\begin{array}{cccccccc} x & y & s_1 & s_2 & s_3 & C & & \end{array} \\
\left[\begin{array}{cccccc:c}
3 & 2 & 1 & 0 & 0 & 0 & 15 \\
-1 & 2 & 0 & 1 & 0 & 0 & 5 \\
1 & 2 & 0 & 0 & 1 & 0 & 7 \\
\hdashline
-2 & -3 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\end{array}
$$

'Pivot' column

Step 2) Identify the column with the **most negative** coefficient in the final row.

- **The Simplex Method (cont.)**

Step 3) Calculate the **row quotients** by dividing each of the entries in the final column by the entries in the pivot column

$$
\begin{array}{c}
\begin{array}{cccccccc} x & y & s_1 & s_2 & s_3 & C & & \end{array} \\
\left[\begin{array}{cccccc:c}
3 & 2 & 1 & 0 & 0 & 0 & 15 \\
-1 & 2 & 0 & 1 & 0 & 0 & 5 \\
1 & 2 & 0 & 0 & 1 & 0 & 7 \\
\hdashline
-2 & -3 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\end{array}
\qquad
\begin{array}{l}
\dfrac{15}{2} = 7.5 \\[1em]
\dfrac{5}{2} = 2.5 \\[1em]
\dfrac{7}{2} = 3.5
\end{array}
$$

Step 4) The row with the **smallest *positive*** row quoteint is the first constraint that is violated when increasing the pivot variable.

Introducing Tableaux

- **The Simplex Method (cont.)**

  Step 5) Apply the **row transformation** that

  - leaves a 1 in the pivot row of the pivot column,
  - leaves a 0 in all other rows of the pivot column

$$
\begin{array}{c}
\begin{array}{cccccccc} x & y & s_1 & s_2 & s_3 & C & & \end{array} \\
\left[\begin{array}{cccccc:c}
3 & 2 & 1 & 0 & 0 & 0 & 15 \\
-1 & 2 & 0 & 1 & 0 & 0 & 5 \\
1 & 2 & 0 & 0 & 1 & 0 & 7 \\
\hdashline
-2 & -3 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\end{array}
$$

$$R_1 \leftarrow (R_1 - R_2) \quad , \quad R_2 \leftarrow \tfrac{1}{2}R_2 \quad , \quad R_3 \leftarrow (R_3 - R_2) \quad , \quad R_4 \leftarrow (R_4 + \tfrac{3}{2}R_2)$$

- **The Simplex Method (cont.)**

  Step 5) Apply the **row transformation** that

  - leaves a 1 in the pivot row of the pivot column,
  - leaves a 0 in all other rows of the pivot column

$$
\begin{array}{cccccc:c}
x & y & s_1 & s_2 & s_3 & C & \\
\end{array}
$$

$$
\left[\begin{array}{cccccc:c}
4 & 0 & 1 & -1 & 0 & 0 & 10 \\
-1 & 2 & 0 & 1 & 0 & 0 & 5 \\
1 & 2 & 0 & 0 & 1 & 0 & 7 \\
\hdashline
-2 & -3 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
$$

$$
R_1 \leftarrow (R_1 - R_2) \quad , \quad R_2 \leftarrow \tfrac{1}{2} R_2 \quad , \quad R_3 \leftarrow (R_3 - R_2) \quad , \quad R_4 \leftarrow (R_4 + \tfrac{3}{2} R_2)
$$

- **The Simplex Method (cont.)**

Step 5) Apply the **row transformation** that

- leaves a 1 in the pivot row of the pivot column,
- leaves a 0 in all other rows of the pivot column

$$
\begin{array}{c}
\begin{array}{cccccccc} x & y & s_1 & s_2 & s_3 & C & & \end{array} \\
\left[\begin{array}{cccccc|c}
4 & 0 & 1 & -1 & 0 & 0 & 10 \\
-1 & 2 & 0 & 1 & 0 & 0 & 5 \\
2 & 0 & 0 & -1 & 1 & 0 & 2 \\
\hline
-2 & -3 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\end{array}
$$

$$R_1 \leftarrow (R_1 - R_2) \quad , \quad R_2 \leftarrow \tfrac{1}{2}R_2 \quad , \quad R_3 \leftarrow (R_3 - R_2) \quad , \quad R_4 \leftarrow (R_4 + \tfrac{3}{2}R_2)$$

- **The Simplex Method (cont.)**

  Step 5) Apply the **row transformation** that

  - leaves a 1 in the pivot row of the pivot column,
  - leaves a 0 in all other rows of the pivot column

$$
\begin{array}{cccccc|c}
x & y & s_1 & s_2 & s_3 & C & \\
\end{array}
$$

$$
\left[\begin{array}{cccccc|c}
4 & 0 & 1 & -1 & 0 & 0 & 10 \\
-1 & 2 & 0 & 1 & 0 & 0 & 5 \\
2 & 0 & 0 & -1 & 1 & 0 & 2 \\
\hline
-3.5 & 0 & 0 & 1.5 & 0 & 1 & 7.5
\end{array}\right]
$$

$$R_1 \leftarrow (R_1 - R_2) \quad , \quad R_2 \leftarrow \tfrac{1}{2}R_2 \quad , \quad R_3 \leftarrow (R_3 - R_2) \quad , \quad R_4 \leftarrow (R_4 + \tfrac{3}{2}R_2)$$

Introducing Tableaux

- **The Simplex Method (cont.)**

  Step 5) Apply the **row transformation** that

  - leaves a 1 in the pivot row of the pivot column,
  - leaves a 0 in all other rows of the pivot column

$$
\begin{array}{cccccc|c}
x & y & s_1 & s_2 & s_3 & C & \\
\hline
4 & 0 & 1 & -1 & 0 & 0 & 10 \\
-0.5 & 1 & 0 & 0.5 & 0 & 0 & 2.5 \\
2 & 0 & 0 & -1 & 1 & 0 & 2 \\
\hline
-3.5 & 0 & 0 & 1.5 & 0 & 1 & 7.5
\end{array}
$$

$$R_1 \leftarrow (R_1 - R_2) \quad , \quad R_2 \leftarrow \tfrac{1}{2}R_2 \quad , \quad R_3 \leftarrow (R_3 - R_2) \quad , \quad R_4 \leftarrow (R_4 + \tfrac{3}{2}R_2)$$

- **The Simplex Method (cont.)**

  Step 5) Repeat until the final row contains only positive coefficients.

$$
\begin{array}{cccccccc}
x & y & s_1 & s_2 & s_3 & C & & \\
\end{array}
$$

$$
\left[\begin{array}{cccccc:c}
4 & 0 & 1 & -1 & 0 & 0 & 10 \\
-0.5 & 1 & 0 & 0.5 & 0 & 0 & 2.5 \\
2 & 0 & 0 & -1 & 1 & 0 & 2 \\
\hdashline
-3.5 & 0 & 0 & 1.5 & 0 & 1 & 7.5
\end{array}\right]
$$

$$R_1 \leftarrow (R_1 - 2R_3) \quad , \quad R_2 \leftarrow (R_2 + \tfrac{1}{4}R_3) \quad , \quad R_3 \leftarrow \tfrac{1}{2}R_3 \quad , \quad R_4 \leftarrow (R_4 + \tfrac{7}{4}R_3)$$

- **The Simplex Method (cont.)**

  Step 5) Repeat until the final row contains only positive coefficients.

$$
\begin{array}{cccccccc}
x & y & s_1 & s_2 & s_3 & C & & \\
\end{array}
$$

$$
\left[\begin{array}{cccccc:c}
0 & 0 & 1 & 1 & -2 & 0 & 6 \\
0 & 1 & 0 & 0.25 & 0.25 & 0 & 3 \\
1 & 0 & 0 & -0.5 & 0.5 & 0 & 1 \\
\hdashline
0 & 0 & 0 & -0.25 & 1.75 & 1 & 11
\end{array}\right]
$$

$$
R_1 \leftarrow (R_1 - 2R_3) \quad , \quad R_2 \leftarrow (R_2 + \tfrac{1}{4}R_3) \quad , \quad R_3 \leftarrow \tfrac{1}{2}R_3 \quad , \quad R_4 \leftarrow (R_4 + \tfrac{7}{4}R_3)
$$

- **The Simplex Method (cont.)**

  Step 5) Repeat until the final row contains only positive coefficients.

$$
\begin{array}{cccccccc}
x & y & s_1 & s_2 & s_3 & C & & \\
\end{array}
$$

$$
\left[\begin{array}{cccccc|c}
0 & 0 & 1 & 1 & -2 & 0 & 6 \\
0 & 1 & 0 & 0.25 & 0.25 & 0 & 3 \\
1 & 0 & 0 & -0.5 & 0.5 & 0 & 1 \\
\hline
0 & 0 & 0 & -0.25 & 1.75 & 1 & 11
\end{array}\right]
$$

$$
\boldsymbol{R}_1 \leftarrow \boldsymbol{R}_1 \quad , \quad \boldsymbol{R}_2 \leftarrow (\boldsymbol{R}_1 - \tfrac{1}{4}\boldsymbol{R}_1) \quad , \quad \boldsymbol{R}_3 \leftarrow (\boldsymbol{R}_3 + \tfrac{1}{2}\boldsymbol{R}_1) \quad , \quad \boldsymbol{R}_4 \leftarrow (\boldsymbol{R}_4 + \tfrac{1}{4}\boldsymbol{R}_1)
$$

- **The Simplex Method (cont.)**

Step 5) Repeat until the final row contains only positive coefficients.

$$
\begin{array}{cccccccc}
 & x & y & s_1 & s_2 & s_3 & C & \\
\end{array}
$$

$$
\left[\begin{array}{cccccc:c}
0 & 0 & 1 & 1 & -2 & 0 & 6 \\
0 & 1 & -0.25 & 0 & 0.75 & 0 & 1.5 \\
1 & 0 & 0.5 & 0 & -0.5 & 0 & 4 \\
\hdashline
0 & 0 & 0.25 & 0 & 1.25 & 1 & 12.5
\end{array}\right]
$$

$$R_1 \leftarrow R_1 \quad , \quad R_2 \leftarrow (R_1 - \tfrac{1}{4}R_1) \quad , \quad R_3 \leftarrow (R_3 + \tfrac{1}{2}R_1) \quad , \quad R_4 \leftarrow (R_4 + \tfrac{1}{4}R_1)$$

- **The Simplex Method (cont.)**

Step 6) Read off the optimal solution

this row tells us the value of s_2

this row tells us the value of y

$$\begin{array}{cccccccc} x & y & s_1 & s_2 & s_3 & C & & \\ \end{array}$$

$$\left[\begin{array}{cccccc:c} 0 & 0 & 1 & 1 & -2 & 0 & 6 \\ 0 & 1 & -0.25 & 0 & 0.75 & 0 & 1.5 \\ 1 & 0 & 0.5 & 0 & -0.5 & 0 & 4 \\ \hdashline 0 & 0 & 0.25 & 0 & 1.25 & 1 & 12.5 \end{array}\right]$$

this row tells us the value of x

$$\boxed{C = 12.5 - 0.25s_1 - 1.25s_3}$$

We can maximise C by setting s_1 and s_3 to be zero

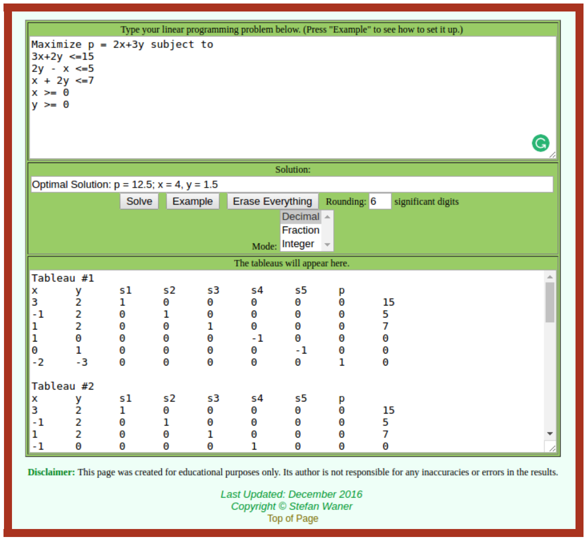
Optimal Solution

$x = 4$ $\quad y = 1.5$ $\quad s_1 = 0$ $\quad s_2 = 6$ $\quad s_3 = 0$ $\quad C = 12.5$

Online Simplex Method Tool

- **Simplex Method Tool**

`http://www.zweigmedia.com/RealWorld/simplex.html`



Branch-and-Bound
for Integer Progamming

Integer Progamming

- **Integer Program**

Linear Program + Require *Integer* solution

  - **Example:**

$$\begin{array}{rrcl} \text{Maximise:} & 2x + 3y & & \\ \text{Subject to:} & 3x + 2y & \leq & 15 \\ & 2y - x & \leq & 5 \\ & x + 2y & \leq & 7 \\ & x,\ y & \geq & 0 \\ & x,\ y & \in & \mathbb{Z} \end{array}$$

(this is the same example as earlier, but with the additional integral requirement)

Integer Progamming

Theorem Integer Programming in NP-complete.

Proof:

Step 1) Consider an instance of the **Knapsack problem**

Item 1	Item 2	Item 3
Weight : 10	Weight : 20	Weight : 30
Value : £ 60	Value : £ 100	Value : £ 120

with Knapsack size : 100

We will express this instance of the Knapsack problem as an Integer Program

Integer Progamming

Step 2) We can express this as an **Integer Program**

Value of Items

Weight of Knapsack

$$
\begin{array}{ll}
\text{Maximise:} & 60x + 100y + 120z \\
\text{Subject to:} & 10x + 20y + 30z \leq 100 \\
& x,\ y,\ z \geq 0 \\
& x,\ y,\ z \in \mathbb{Z}
\end{array}
$$

Non-negativity constraint

Integral constraint

Since Knapsack is NP-complete, so too must be Integer Programming.

Q.E.D

- **Branch-and-Bound Algorithm**

Step 1) Solve the **'continuous/linear relaxation'** using the Simplex Method

(remove the requirement that solution must be integers)

$$
\begin{array}{lrcl}
\text{Maximise:} & 2x + 3y & & \\
\text{Subject to:} & 3x + 2y & \leq & 15 \\
 & 2y - x & \leq & 5 \\
 & x + 2y & \leq & 7 \\
 & x,\ y & \geq & 0 \\
 & \cancel{x,\ y} & \cancel{\in} & \cancel{\mathbb{Z}}
\end{array}
$$

Solution

$x = 4$ and $y = 1.5$

Step 2) If all values are integral then we are done!

We can just **return the solution** that we have found!

- **Branch-and-Bound Algorithm (cont.)**

Step 3) Else, **branch** to two sub-problems with the variable **bounded** above and below by the floor and ceiling of the previous solution

$$\begin{array}{rrcl} \text{Maximise:} & 2x+3y & & \\ \text{Subject to:} & 3x+2y & \leq & 15 \\ & 2y-x & \leq & 5 \\ & x+2y & \leq & 7 \\ & x,y & \geq & 0 \\ & y & \leq & 1 \end{array}$$

$$\begin{array}{rrcl} \text{Maximise:} & 2x+3y & & \\ \text{Subject to:} & 3x+2y & \leq & 15 \\ & 2y-x & \leq & 5 \\ & x+2y & \leq & 7 \\ & x,y & \geq & 0 \\ & y & \geq & 2 \end{array}$$

Additional Constraints

Step 4) Recursively apply Branch-and-Bound to both sub-problems and return solution which **maximises** the objective function.

(this is yet another example of *Divide-and-Conquer*)

Branch-and-Bound

- **Overview**

$$\begin{array}{lrcl} \text{Maximise:} & 2x+3y & & \\ \text{Subject to:} & 3x+2y & \leq & 15 \\ & 2y-x & \leq & 5 \\ & x+2y & \leq & 7 \\ & x,y & \geq & 0 \end{array}$$

$$\boxed{x = 4, \quad y = 1.5}$$

Cost = 12.5

$$\begin{array}{lrcl} \text{Maximise:} & 2x+3y & & \\ \text{Subject to:} & 3x+2y & \leq & 15 \\ & 2y-x & \leq & 5 \\ & x+2y & \leq & 7 \\ & x,y & \geq & 0 \\ & y & \leq & 1 \end{array}$$

$$\boxed{x = 4.3, \quad y = 1}$$

Cost = 11.6

$$\begin{array}{lrcl} \text{Maximise:} & 2x+3y & & \\ \text{Subject to:} & 3x+2y & \leq & 15 \\ & 2y-x & \leq & 5 \\ & x+2y & \leq & 7 \\ & x,y & \geq & 0 \\ & y & \geq & 2 \end{array}$$

$$\boxed{x = 3, \quad y = 2}$$

Cost = 12

$$\begin{array}{lrcl} \text{Maximise:} & 2x+3y & & \\ \text{Subject to:} & 3x+2y & \leq & 15 \\ & 2y-x & \leq & 5 \\ & x+2y & \leq & 7 \\ & x,y & \geq & 0 \\ & y & \leq & 1 \\ & x & \leq & 4 \end{array}$$

$$\boxed{x = 4, \quad y = 1}$$

Cost = 11

$$\begin{array}{lrcl} \text{Maximise:} & 2x+3y & & \\ \text{Subject to:} & 3x+2y & \leq & 15 \\ & 2y-x & \leq & 5 \\ & x+2y & \leq & 7 \\ & x,y & \geq & 0 \\ & y & \leq & 1 \\ & x & \geq & 5 \end{array}$$

$$\boxed{x = 5, \quad y = 0}$$

Cost = 10

Branch-and-Bound

- Hence the solution to our **Integer Program** is

Solution

$x = 3$ and $y = 2$

$\rightsquigarrow$ **Cost = 12**

- The solution to the Integer Program is always **less optimal** than the solution to its **continuous/linear relaxation**.

- We can **cut corners** by only branching on those children whose are **no worse** than the best integer solution found so far!

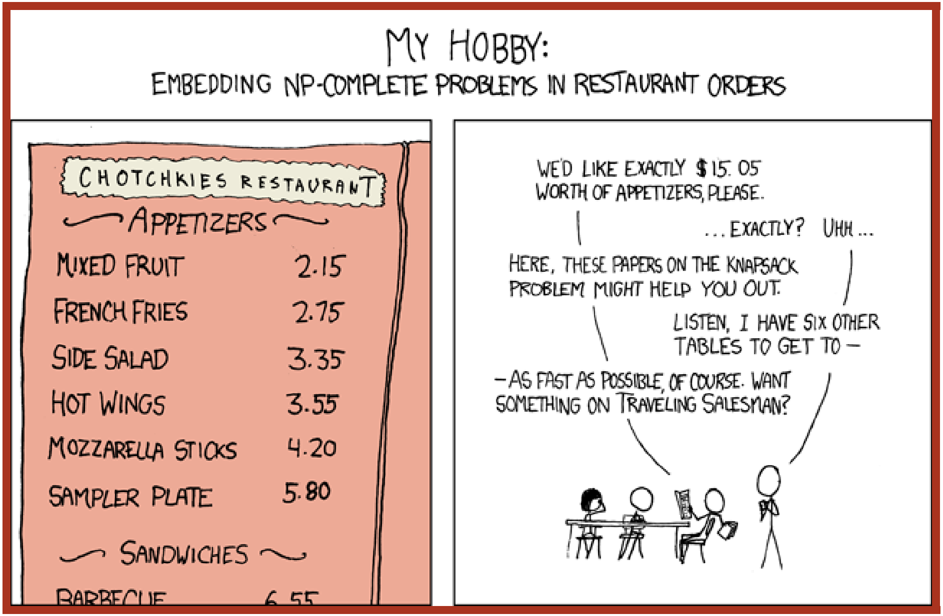
(if we had evaluated the right-child before the left-child, we could have stopped early since $11.6 < 12$)

Other Variants of Linear Programming

- **Mixed Integer Linear Progamming (MILP)**
  - Hybrid of an **Integer Program** and a classical **Linear Program**,
  - Some variable are required to be **integers**,
  - Others are allowed to take **non-integral** values.

- **Zero-one Integer Programming**
  - A restriction of **Integer Progamming**,
  - All variables can be either **zero** or **one**,

Other Variants of Linear Programming



https://xkcd.com/287/

Next Time...

- **Week 9**
  - Approximate Algorithms,
  - Approximation Ratio and Unapproximable Problems,

End of Slides!

Feedback

- **Let me know how you found today's lecture?**

https://goo.gl/forms/aW5x0iw0Doiq09Mt2