Algorithms and Data Structures

Degeneracy, Geometry, and Duality

What about this dictionary?

$$
\begin{array}{llrrrr}
\textbf{Maximise} & \zeta = & 3 & -0.5\,x_1 & +2\,x_2 & -1.5\,w_1 \\
\textbf{subject to} & x_3 = & 1 & -0.5\,x_1 & & -0.5\,w_1 \\
 & w_2 = & & x_1 & -\,x_2 & +\,w_1
\end{array}
$$

$$x_1, x_2, x_3, w_1, w_2 \geq 0$$

leaving variable

entering variable

We can increase the value of some nonbasic variable, here x_2

We should not violate any constraints though!

We don't want any of the slack variables to become negative.

x_2 cannot be increased! Are we stuck?

Degeneracy!

Degeneracy

Degenerate dictionary: A dictionary in which one of the b_i variables becomes zero.

Equivalently: In a basic feasible solution, one of the basic variables is 0.

Degeneracy not necessarily and issue

entering variable

Maximise $\zeta = 5 + x_3 - x_1$

subject to

$$\begin{aligned} x_2 &= 5 + 2x_3 - 3x_1 \\ x_4 &= 7 - 4x_1 \\ x_5 &= x_1 \end{aligned}$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

The LP is unbounded!

We can increase the value of some nonbasic variable, here x_3

We should not violate any constraints though!

We don't want any of the slack variables to become negative.

This does not happen regardless of how much we increase x_3.

Degeneracy

Degenerate dictionary: A dictionary in which one of the b_i variables becomes zero.

Equivalently: In a basic feasible solution, one of the basic variables is 0.

Degenerate Pivot: The entering variable stays at 0 without increasing.

What about this dictionary?

$$
\begin{array}{llrrrr}
\textbf{Maximise} & \zeta = & 3 & -0.5\,x_1 & +2\;x_2 & -1.5\,w_1 \\
\\
\textbf{subject to} & x_3 = & 1 & -0.5\;x_1 & & -0.5\;w_1 \\
 & w_2 = & & x_1 & -\;x_2 & +\;w_1
\end{array}
$$

entering variable

leaving variable

$$x_1, x_2, x_3, w_1, w_2 \geq 0$$

We can increase the value of some nonbasic variable, here x_2

We should not violate any constraints though!

We don't want any of the slack variables to become negative.

x_2 cannot be increased! Are we stuck?

Degeneracy!

Degeneracy

Degenerate dictionary: A dictionary in which one of the b_i variables becomes zero.

Equivalently: In a basic feasible solution, one of the basic variables is 0.

Degenerate Pivot: The entering variable stays at 0 without increasing.

"Degenerate pivots are quite common and usually harmless."

Let's not give up

Maximise $\zeta = 3 - 0.5\,x_1 + 2\,x_2 - 1.5\,w_1$

subject to

$$x_3 = 1 - 0.5\,x_1 - 0.5\,w_1$$

leaving variable

$$w_2 = x_1 - x_2 + w_1$$

entering variable

$$x_1, x_2, x_3, w_1, w_2 \geq 0$$

We can increase the value of some nonbasic variable, here x_2

We should not violate any constraints though!

We don't want any of the slack variables to become negative.

x_2 cannot be increased! Are we stuck?

Let’s not give up

Maximise $\zeta = 3 - 0.5\,x_1 + 2\,x_2 - 1.5\,w_1$

subject to

$$
\begin{aligned}
x_3 &= 1 - 0.5\,x_1 - 0.5\,w_1 \\
w_2 &= x_1 - x_2 + w_1
\end{aligned}
$$

leaving variable

entering variable

$$x_1, x_2, x_3, w_1, w_2 \geq 0$$

Actually pivot!

$(x_1, x_2, x_3, w_1, w_2) = (0,0,1,0,0)$

We can increase the value of some nonbasic variable, here x_2

We should not violate any constraints though!

We don’t want any of the slack variables to become negative.

Increase the variable as much as we can (as before): here 0 increase

The new dictionary

entering variable

$$
\begin{array}{llllll}
\textbf{Maximise} & \zeta = & 3 & +1.5\, x_1 & +2\, w_2 & +0.5\, w_1 \\
\textbf{subject to} & x_3 = & 1 & -0.5\, x_1 & & -0.5\, w_1 \\
 & x_2 = & & x_1 & -\, w_2 & +\, w_1
\end{array}
$$

leaving variable

$$x_1, x_2, x_3, w_1, w_2 \geq 0$$

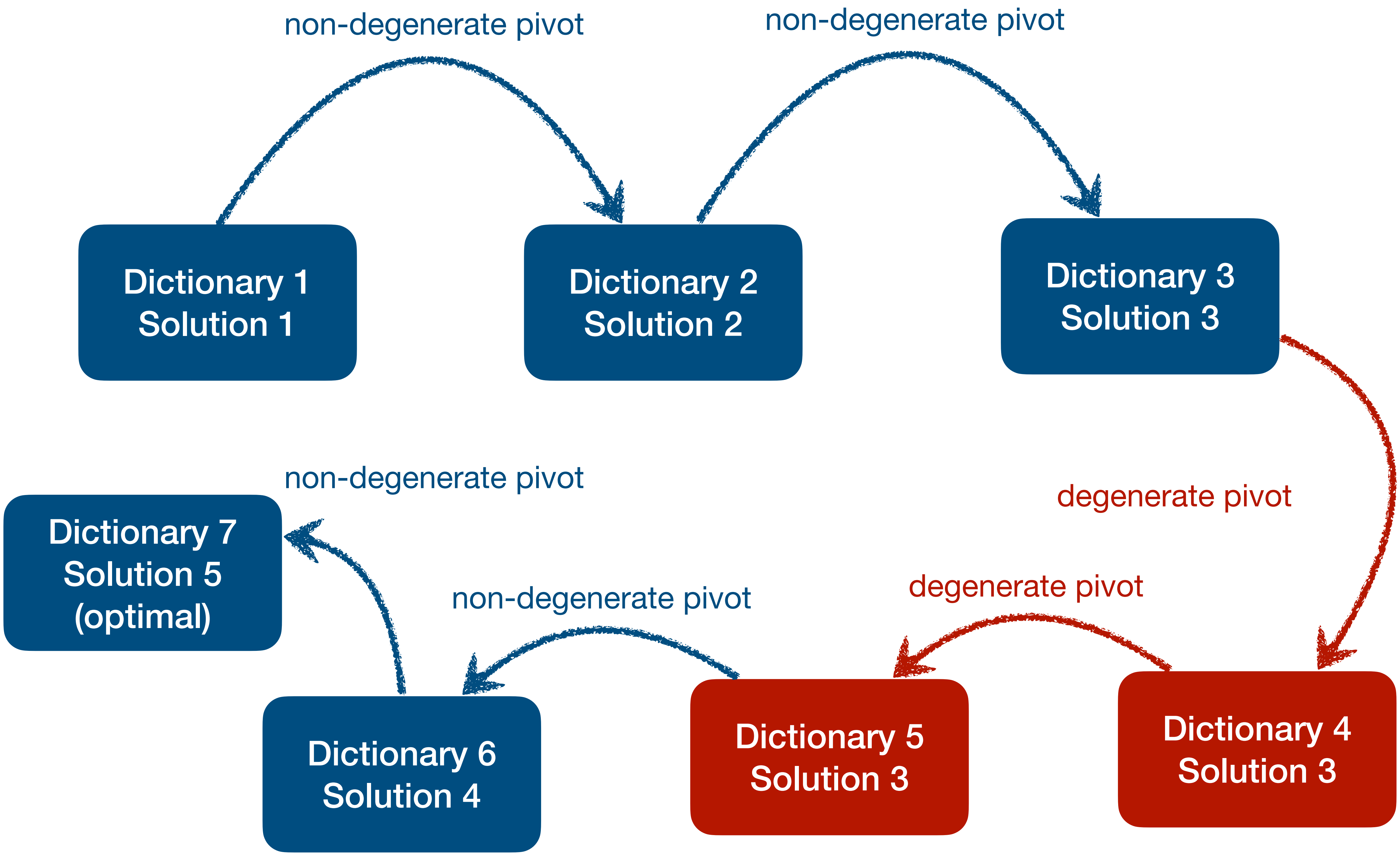
$$(x_1, x_2, x_3, w_1, w_2) = (0,0,1,0,0)$$

We can now increase x_1 to $x_1 = 2$

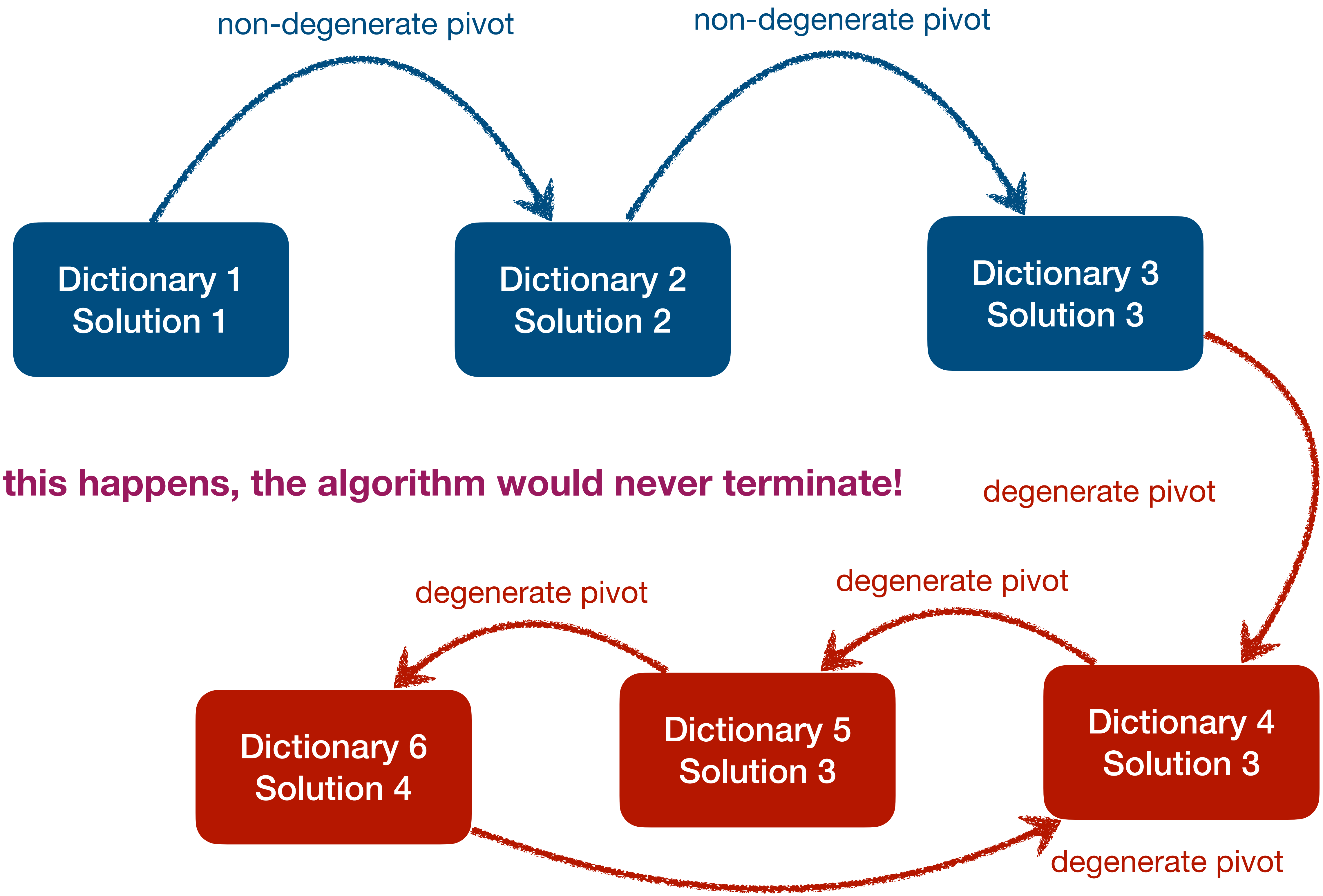
The pivot is not degenerate!

It will actually lead to a final dictionary, and an optimal solution.

Pictorially



Could this happen though?



If this happens, the algorithm would never terminate!

Cycling

In theory: Cycling can happen.

In practice: Cycling rarely happens.

But non-degenerate pivots are quite common.

Can we avoid cycling in theory too?

Bland's rule: For both the entering variable and the leaving variable, choose the one with the smallest index.

Termination

Theorem: If the simplex method does not cycle, it terminates.

Proof: A dictionary is determined by which variables are basic and which are non-basic.

There only $\binom{n+m}{m} = \dfrac{(n+m)!}{n!m!}$ possibilities.

Termination

Theorem: If the simplex method does not cycle, it terminates.

Proof: A dictionary is determined by which variables are basic and which are non-basic.

There only $\binom{n+m}{m} = \frac{(n+m)!}{n!m!}$ possibilities.

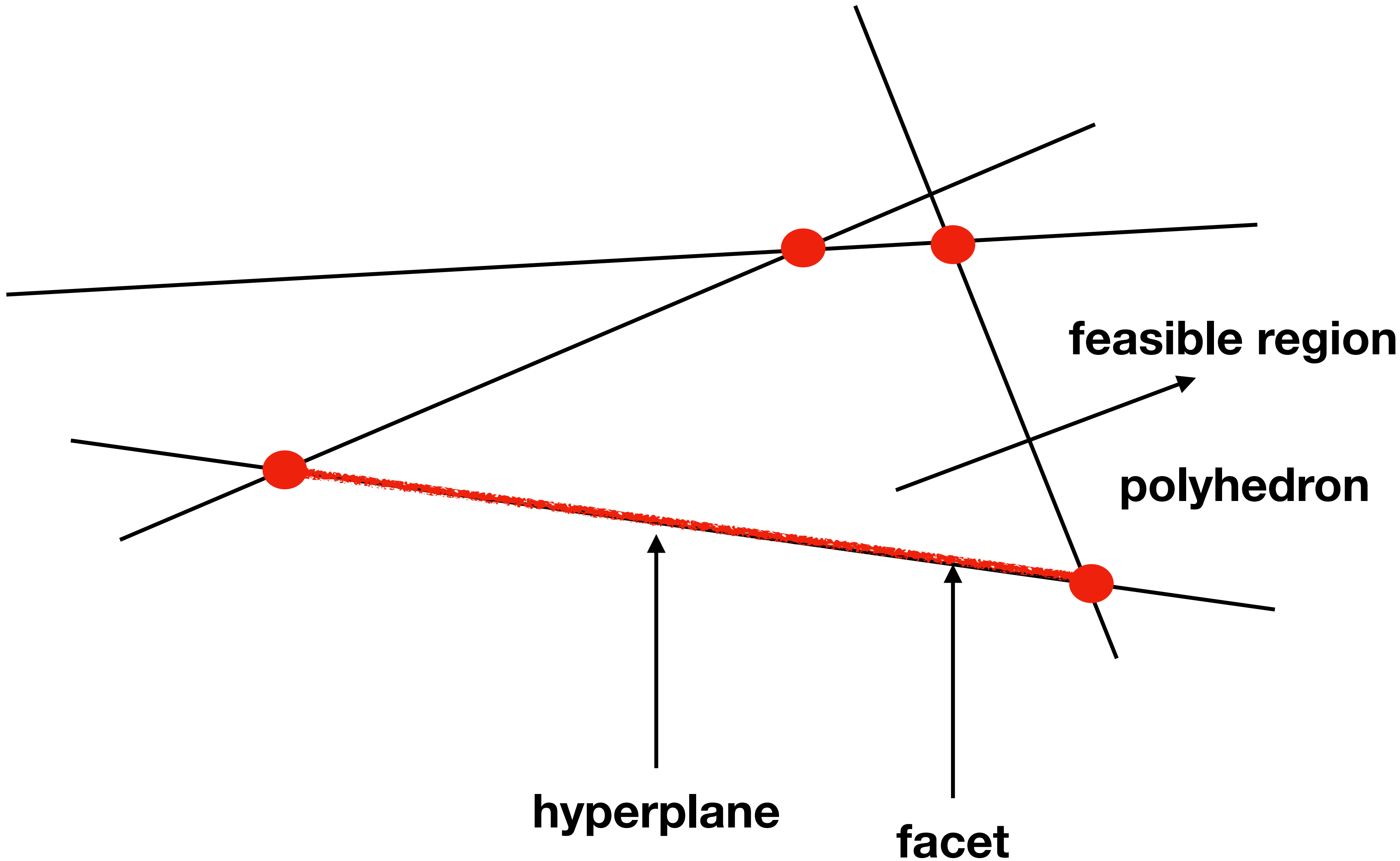
Geometry

Every constraint corresponds to a hyperplane, which defines a halfspace.

The intersection of those halfspaces is the feasible region, which is a polyhedron (or polytope).

The part of each hyperplane that intersects with the feasible region is called a facet.

Geometry



Geometry

Every constraint corresponds to a hyperplane, which defines a halfspace.

The intersection of those halfspaces is the feasible region, which is a polyhedron (or polytope).

The part of each hyperplane that intersects with the feasible region is called a facet.

A facet corresponds to a constraint satisfied with equality, e.g., $x_1 + 2x_3 = 3$

In terms of the dictionary, a facet corresponds to the corresponding variable (original or slack) being 0.

Geometry

Consider an LP with three variables x_1, x_2, x_3.

In terms of the dictionary, a facet corresponds to the corresponding variable (original or slack) being 0.

An edge corresponds to two variables being 0.

A vertex corresponds to three variables being 0.

Geometry

Maximise $5x_1 + 4x_2 + 3x_3$

subject to

$$\left.\begin{aligned} 2x_1 + 3x_2 + x_3 &\le 5 \\ 4x_1 + x_2 + 2x + 3 &\le 11 \\ 3x_1 + 4x_2 + 2x_3 &\le 8 \end{aligned}\right\}$$

$w_1 = w_2 = w_3 = 0$
corresponds to the intersection of
these three hyperplanes.

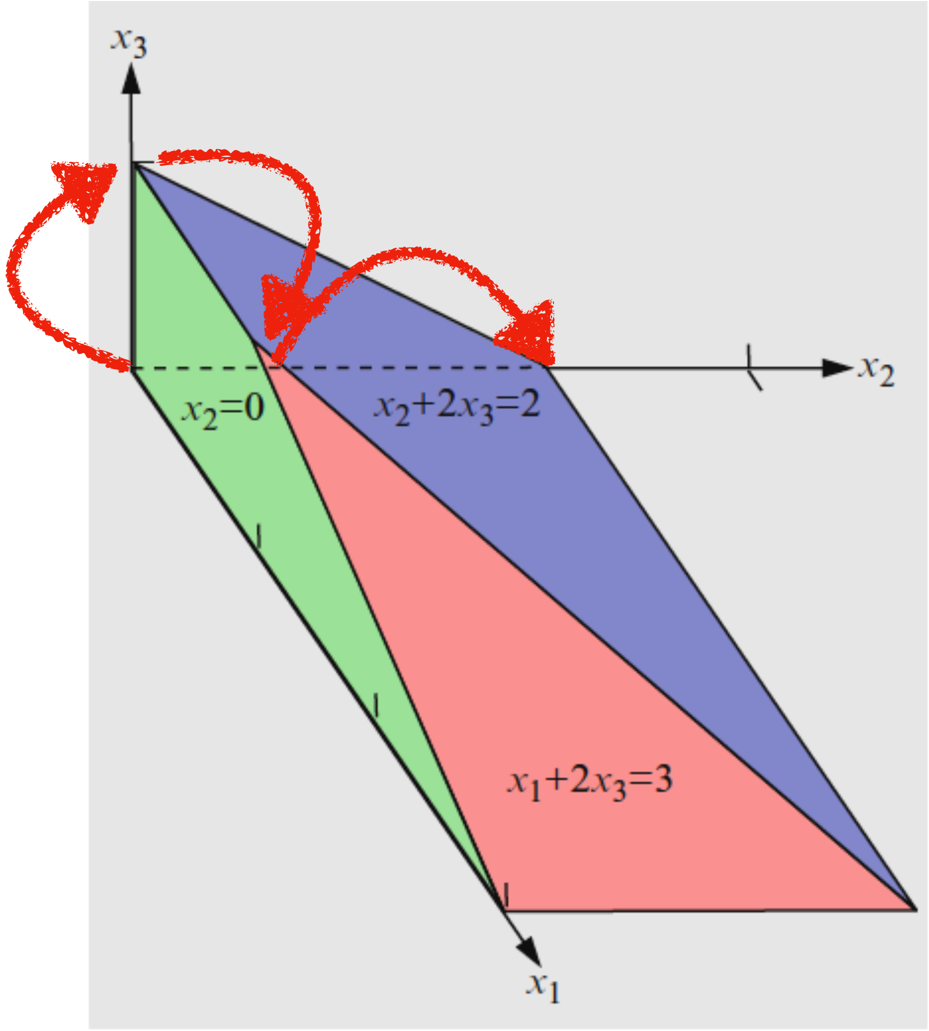
$$x_1, x_2, x_3 \ge 0$$

Example

Maximise $x_1 + 2x_2 + 3x_3$

subject to

$$x_1 + 2x_3 \leq 3$$
$$x_2 + 2x_3 \leq 2$$
$$x_1, x_2, x_3 \geq 0$$



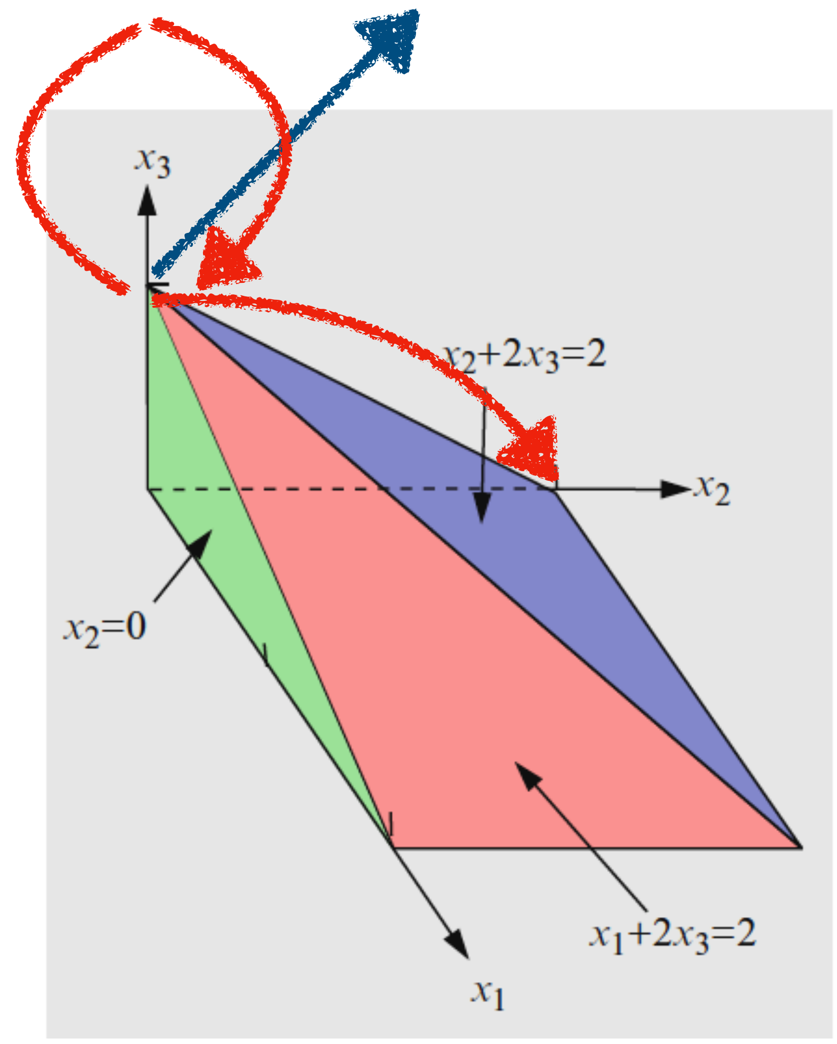
Example

Maximise $x_1 + 2x_2 + 3x_3$

subject to

$$x_1 + 2x_3 \leq 2$$
$$x_2 + 2x_3 \leq 2$$
$$x_1, x_2, x_3 \geq 0$$

intersection of four facets



The intersection point corresponds to the same solution of the LP.

But it corresponds to four different basic feasible solutions/dictionaries.

Termination

Theorem: If the simplex method does not cycle, it terminates.

Proof: A dictionary is determined by which variables are basic and which are non-basic.

There only $\binom{n+m}{m} = \frac{(n+m)!}{n!m!}$ possibilities.

Simplex Running Time

First: Simplex is a method, not an algorithm, parameterised by the pivoting rule.

Bad news: None of the known pivoting rules are known to result in a polynomial-time algorithm.

More bad news: Most of the known pivoting rules have been shown to result in exponential running time.

Good news: In practice the algorithm is quite efficient/fast.

More good news: "*Beyond the worst-case analysis*" shows that the algorithm is also efficient in theory.

Even more good news: We have other algorithms that run in worst-case polynomial running time (Ellipsoid Method, Interior Point Methods).

Duality

Suppose that we have a linear program, which we will refer to as *the primal*.

We will construct another linear program, which we will refer to as *the dual*.

The *variables* of the primal become the *constraints* of the dual and vice-versa.

Maximisation becomes minimisation.

The two linear programs will have a very important connection.

The Primal

Maximise

$c_1x_1 + c_2x_2 + \ldots + c_nx_n$

Subject to

$a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n \leq b_1$

$a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n \leq b_2$

$\vdots$

$a_{m,1}x_1 + a_{m,2}x_2 + \ldots + a_{m,n}x_n \leq b_m$

$x_1, \ldots, x_n \geq 0$

Maximise $c^\top x$

Subject to

$Ax \leq b$

$x_1, \ldots, x_n \geq 0$

The Dual

Minimise

$b_1 y_1 + b_2 y_2 + \ldots + b_m y_m$

Subject to

$a_{1,1}y_1 + a_{1,2}y_2 + \ldots + a_{1,m}y_m \geq c_1$

$a_{2,1}y_1 + a_{2,2}y_2 + \ldots + a_{2,m}y_m \geq c_2$

$\vdots$

$a_{n,1}y_1 + a_{n,2}y_2 + \ldots + a_{n,m}y_m \geq c_n$

$y_1, \ldots, y_m \geq 0$

Minimise $b^\top y$

Subject to

$A^\top y \geq c$

$y_1, \ldots, y_m \geq 0$

Side by side

n variables
m constraints

Maximise

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

Subject to

$$a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n \leq b_1$$
$$a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n \leq b_2$$
$$\vdots$$
$$a_{m,1}x_1 + a_{m,2}x_2 + \ldots + a_{m,n}x_n \leq b_m$$
$$x_1, \ldots, x_n \geq 0$$

$\leq$ becomes $\geq$

m variables
n constraints

Minimise

$$b_1y_1 + b_2y_2 + \ldots + b_my_m$$

Subject to

$$a_{1,1}y_1 + a_{1,2}y_2 + \ldots + a_{1,m}y_m \geq c_1$$
$$a_{2,1}y_1 + a_{2,2}y_2 + \ldots + a_{2,m}y_m \geq c_2$$
$$\vdots$$
$$a_{n,1}y_1 + a_{n,2}y_2 + \ldots + a_{n,m}y_m \geq c_n$$
$$y_1, \ldots, y_m \geq 0$$

Matrix A gets transposed

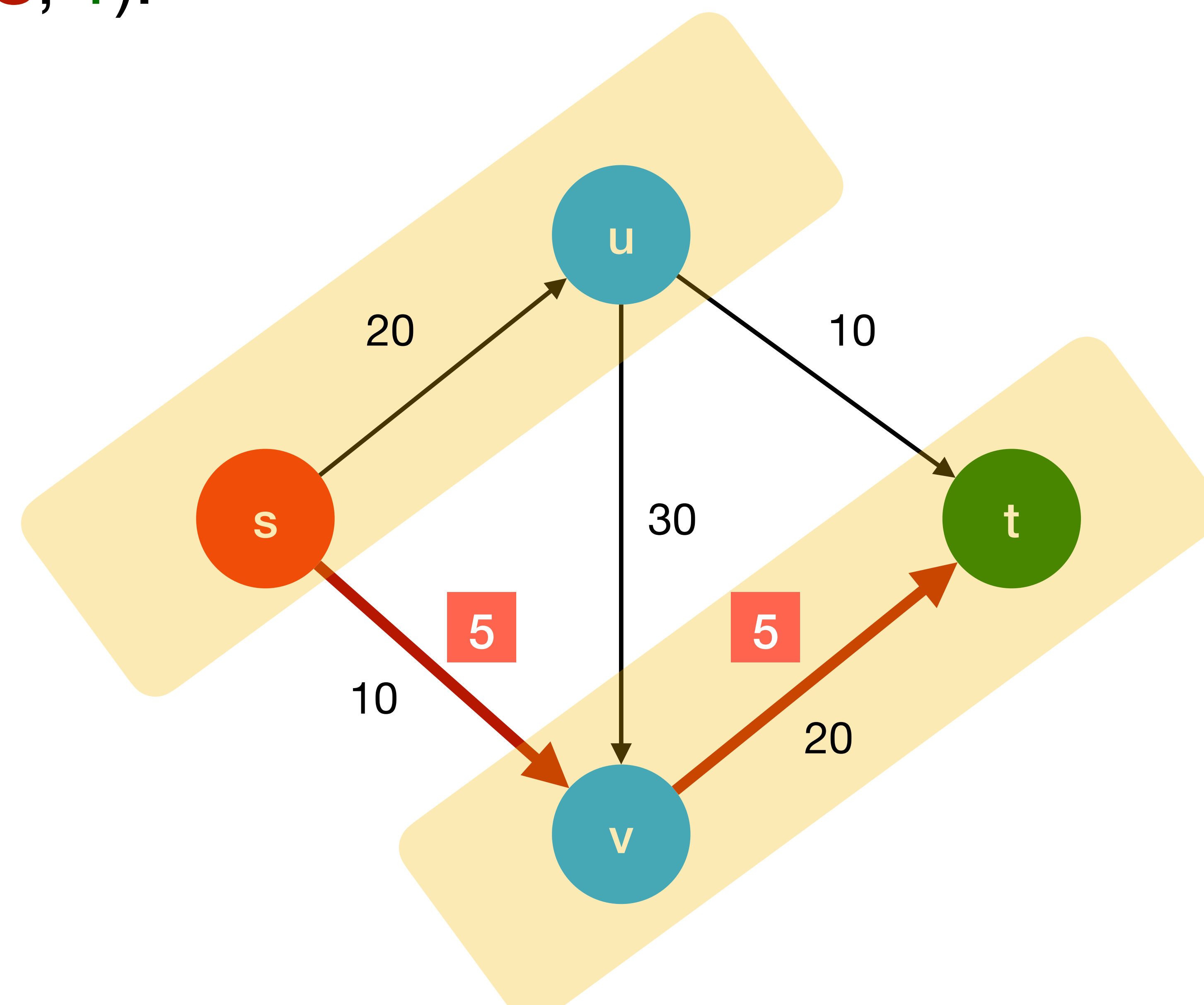
Weak Duality

Let x be any feasible solution to the Primal and let y be any feasible solution to the Dual. Then we have that

$$\text{value}(x) \leq \text{value}(y)$$

Weak Duality

Fact 3: Let f by any (s-t) flow and (S, T) be any (s-t) cut. Then v(f) ≤ c(S, T).



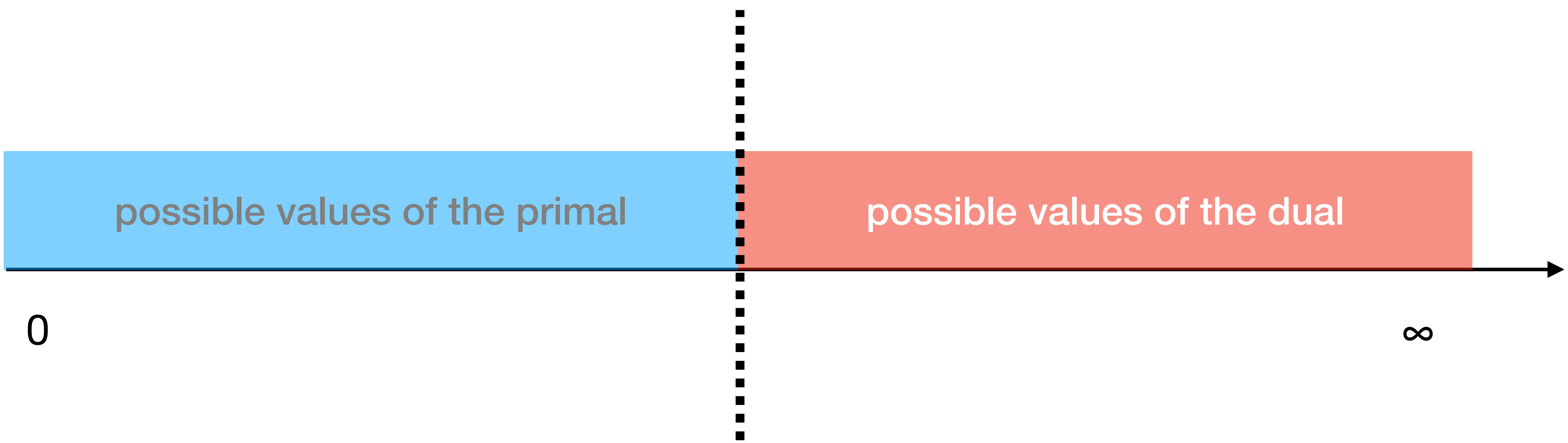
Strong Duality

Let x be any feasible solution to the Primal and let y be any feasible solution to the Dual. If

$$\text{value}(x) = \text{value}(y)$$

then x and y are both optimal solutions.

Pictorially



How can we prove that a solution x to the primal is maximum?

Find a solution y to the dual with value(y) = value(x)

Strong Duality (complete statement)

Theorem (Strong Duality, von Neumann 1947): One of the following is true:

1. Both the primal and the dual are feasible, and let $x^* \in \mathbb{R}^n$ and $y^* \in \mathbb{R}^m$ be any optimal solutions to the primal and the dual, respectively. Then $c^\top x^* = b^\top y^*$.

2. The primal is infeasible and the dual is unbounded.

3. The primal is unbounded and the dual is infeasible.

4. Both the primal and the dual are infeasible.

LP-Duality and the Simplex Method

The proof of the strong duality theorem follows from the proof of correctness of the Simplex method.

We will not cover this, take the strong duality theorem as a given.

Interesting observation: Consider the final dictionary of the simplex method at which we compute an optimal solution x^* to the primal.

Let $y_j^* = -\hat{c}_{n+j}$, where $\hat{c}_{n+j}$ is the coefficient of the slack variable x_{n+j}.

The values y^* obtained that way are an optimal solution to the dual!

The Final Dictionary

$-y_1^*$

Maximise $\zeta = 13 - w_1 - 2x_2 - w_3$

$-y_3^*$

subject to

$$
\begin{aligned}
x_1 &= 2 - 2w_1 - 2x_2 + w_3 \\
w_2 &= 1 + 2w_1 + 5x_2 \\
x_3 &= 1 + 3w_1 + x_2 - 2w_3
\end{aligned}
$$

$$x_1, x_2, x_3, w_1, w_2, w_3 \geq 0$$

$$w_1 = 0,\ x_2 = 0\ w_3 = 0 \qquad x_1 = 2,\ w_2 = 1,\ x_3 = 1$$

Sanity check

Maximise $5x_1 + 4x_2 + 3x_3$

$y^* = (1,0,1)$

subject to

$$2x_1 + 3x_2 + x_3 \leq 5$$
$$4x_1 + x_2 + 2x + 3 \leq 11$$
$$3x_1 + 4x_2 + 2x_3 \leq 8$$
$$x_1, x_2, x_3 \geq 0$$

Minimise $\underline{5y_1 + 11y_2 + 8y_3}$

$5 * 1 + 8 * 1 = 13$

subject to

$$2y_1 + 4y_2 + 3y_3 \geq 5$$
$$3y_1 + y_2 + 4y_3 \geq 4$$
$$y_1 + 2y_2 + 2y_3 \geq 3$$
$$y_1, y_2, y_3 \geq 0$$

Complementary Slackness

Proposition (Complementary Slackness): Let x^* and y^* be feasible solutions to the primal and the dual respectively. Then x^* and y^* are both optimal if and only if both of the following hold:

- For each $i = 1, \ldots, m$, we have $\left((Ax^*)_i - b_i\right) \cdot y_i^* = 0$

- For each $j = 1, \ldots, n$, we have $\left((A^\top y^*)_j - c_j\right) \cdot x_j^* = 0$

On an example

Maximise $5x_1 + 4x_2 + 3x_3$

$x^* = (2,0,1)$

$y^* = (1,0,1)$

subject to

$2x_1 + 3x_2 + x_3 \leq 5$ $\quad 2*2+3*0+2*1=5$

$4x_1 + x_2 + 2x + 3 \leq 11$

$3x_1 + 4x_2 + 2x_3 \leq 8$

$x_1, x_2, x_3 \geq 0$

$x_1^* > 0$

$x_2^* = 0$

$x_3^* > 0$

Minimise $5y_1 + 11y_2 + 8y_3$

$y_1^* > 0$

subject to

$2*1+4*0+3*1=5$ $\quad 2y_1 + 4y_2 + 3y_3 \geq 5$

$3*1+0+4*1=7>5$ $\quad 3y_1 + y_2 + 4y_3 \geq 4$

$1*1+2*0+2*1=3$ $\quad y_1 + 2y_2 + 2y_3 \geq 3$

$y_1, y_2, y_3 \geq 0$

Complementary Slackness

Proposition (Complementary Slackness): Let x^* and y^*be feasible solutions to the primal and the dual respectively. Then x^* and y^* are both optimal if and only if both of the following hold:

- For each $i = 1, \ldots, m$, we have $\left((Ax^*)_i - b_i\right) \cdot y_i^* = 0$
- For each $j = 1, \ldots, n$, we have $\left((A^\top y^*)_j - c_j\right) \cdot x_j^* = 0$

Complementary Slackness

Proof:

We know that $c^\top x^* \le \left((y^*)^\top A\right) x^* = (y^*)^\top (Ax^*) \le (y^*)^\top b$
by weak duality.

By strong duality, when both x^* and y^* are optimal, the LHS is equal to the RHS, which means that all inequalities hold with equality.

So in that case we have: $\left(((y^*)^\top A) - c^\top\right) x^* = 0$.

This is only possible if the product is zero for each coordinate, since both terms are non-negative.

So for each $j = 1, \ldots, n$, we have $\left((A^\top y^*)_j - c_j\right) \cdot x_j^* = 0$.

Similarly for the case of $\left((Ax^*)_i - b_i\right) \cdot y_i^* = 0$

The Max-Flow Min-Cut Theorem

Theorem: In every flow network, the value of the maximum flow is *equal* to the capacity of the minimum cut.

This is a consequence of the *strong duality theorem* for linear programs!

Maximum Flow as an LP

We can write the maximum flow problem as a linear problem.

"Maximise the flow, subject to capacity and flow conservation constraints".

total flow out of s　　total flow into s

capacity constraint

flow conservation

non-negative flow

$$
\begin{array}{lll}
\text{maximise} & \displaystyle\sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} & \\
\text{subject to} & f_{uv} \le c_{uv}, & \text{for each } u, v \in V \\
& \displaystyle\sum_{v\in V} f_{vu} = \sum_{v\in V} f_{uv}, & \text{for each } u \in V - \{s, t\} \\
& f_{uv} \ge 0, & \text{for each } u, v \in V
\end{array}
$$

Constructing the dual

This is *1* if u is in S and v is in T
and *0* otherwise.

This is *1* if u is in S and *0* otherwise.

$$
\begin{aligned}
\text{minimise} \quad & \sum_{(u,v)\in E} c_{uv}\boxed{d_{uv}} \\
\text{subject to} \quad & d_{uv} - \boxed{z_u} + z_v \geq 0 \quad \text{for each } (u,v) \in E, u \neq s, v \neq t \\
& d_{su} + z_v \geq 1 \quad \text{for each } (s,u) \in E \\
& d_{ut} - z_u \geq 0 \quad \text{for each } (u,t) \in E \\
& d_{uv} \geq 0, \quad \text{for each } (u,v) \in E \\
& z_u \geq 0 \quad \text{for each } u \in V - \{t,s\} \\
& d_{uv} \in \{0,1\}, \quad \text{for each } (u,v) \in E \\
& z_u \in \{0,1\}, \quad \text{for each } u \in V - \{s,t\}
\end{aligned}
$$

If u is in S and v is in T, then d_{uv} must be *1*.

If v is in T then d_{sv} must be *1*.

If u is in S then d_{ut} must be *1*.

Minimum Cut as an ILP

$$
\begin{array}{lrll}
\text{minimise} & \displaystyle\sum_{(u,v)\in E} c_{uv} d_{uv} & & \\
\text{subject to} & d_{uv} - z_u + z_v \geq 0 & \text{for each } (u,v) \in E, u \neq s, v \neq t & \\
& d_{su} + z_v \geq 1 & \text{for each} & (s,u) \in E \\
& d_{ut} - z_u \geq 0 & \text{for each} & (u,t) \in E \\
& d_{uv} \geq 0, & \text{for each} & (u,v) \in E \\
& z_u \geq 0 & \text{for each } u \in V - \{t,s\} & \\
& d_{uv} \in \{0,1\}, & \text{for each } (u,v) \in E & \\
& z_u \in \{0,1\}, & \text{for each } u \in V - \{s,t\} &
\end{array}
$$

LP-relaxation

An *LP-relaxation* of an Integer Linear Program is a linear program which is identical to the ILP, except all the integrality constraints have been removed ("*relaxed*"), or replaced with non-integral constraints.

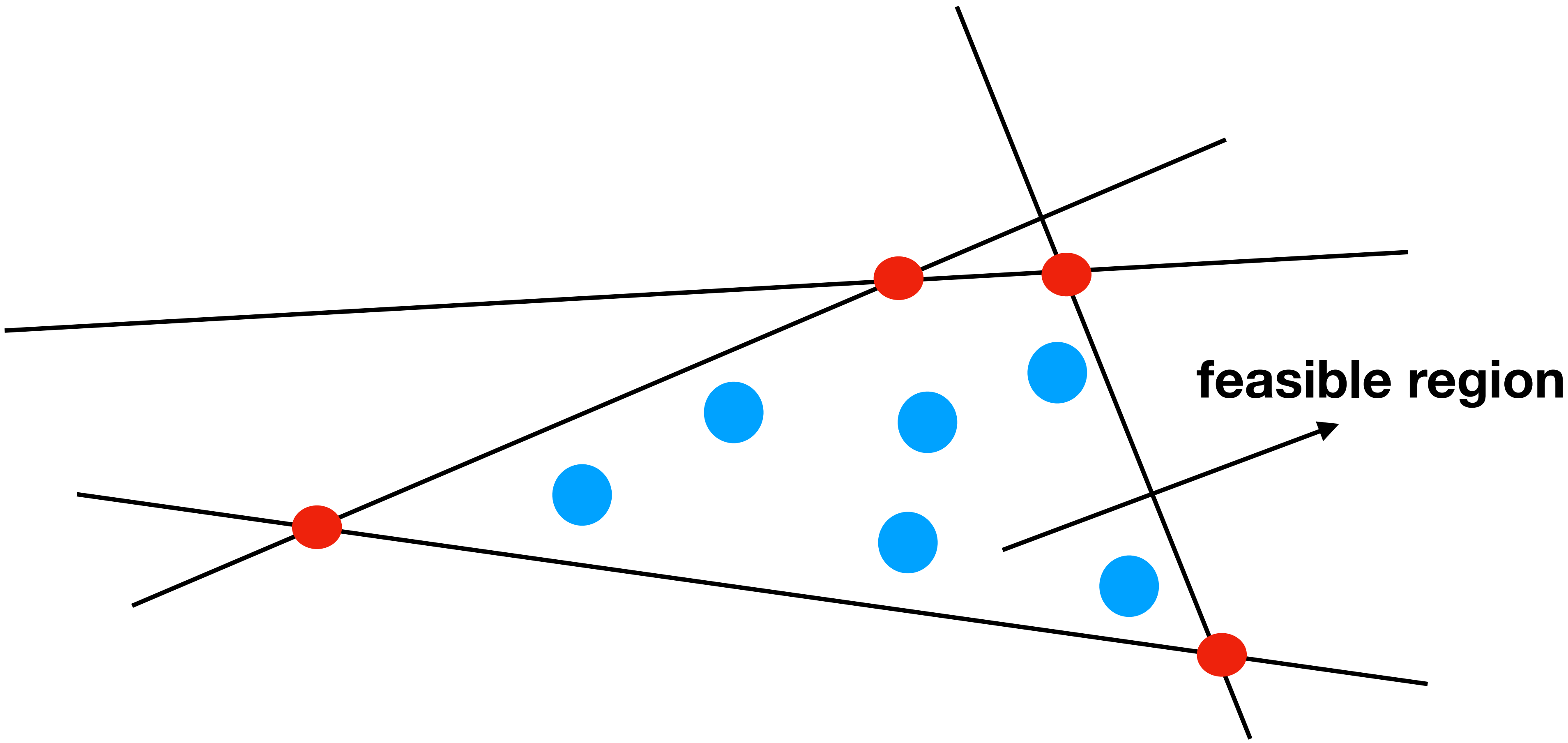
Minimum Cut as an ILP

LP-relaxation

$$
\begin{array}{lll}
\text{minimise} & \displaystyle\sum_{(u,v)\in E} c_{uv} d_{uv} & \\
\text{subject to} & d_{uv} - z_u + z_v \geq 0 & \text{for each } (u,v) \in E, u \neq s, v \neq t \\
 & d_{su} + z_v \geq 1 & \text{for each } \qquad (s,u) \in E \\
 & d_{ut} - z_u \geq 0 & \text{for each } \qquad (u,t) \in E \\
 & d_{uv} \geq 0, & \text{for each } \qquad (u,v) \in E \\
 & z_u \geq 0 & \text{for each } u \in V - \{t,s\}
\end{array}
$$

$$d_{uv} \in \{0,1\}, \quad \text{for each } (u,v) \in E$$
$$z_u \in \{0,1\}, \quad \text{for each } u \in V - \{s,t\}$$

ILP vs LP-relaxation



candidate optimal solution for ILP

candidate optimal solution for LP-relaxation

ILP vs LP-relaxation

For a maximisation problem:

The optimal value of the ILP is not larger than the optimal value of the LP-relaxation.

The ratio

max_value(LP-relaxation) / max_value(LP)

is called the integrality gap of the LP-formulation.

Let’s put some facts together.

The Max-Flow LP and the Min-Cut LP-relaxation are duals of each other.

By *strong duality*, it holds that the optimal value of the Max-Flow LP is equal to the optimal value of the Min-Cut LP-relaxation.

By the *Max-Flow-Min-Cut Theorem,* the value of the maximum flow is *equal* to the capacity of the minimum cut.

That can only mean one thing:

The value of the Min-Cut LP-relaxation is equal to the value of the Min-Cut LP.

In other words, the Min-Cut LP-formulation has integrality gap *1*.

In other words, the Min-Cut LP has *an integer optimal solution*.

Back to Maximum Flow

What if we wanted an integer flow instead of any flow?

$$
\begin{array}{lrll}
\text{maximise} & \displaystyle\sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} & & \\
\text{subject to} & f_{uv} \le c_{uv}, & \text{for each} & u, v \in V \\
& \displaystyle\sum_{v \in V} f_{vu} = \sum_{v \in V} f_{uv}, & \text{for each } u \in V - \{s, t\} & \\
& f_{uv} \ge 0, & \text{for each} & u, v \in V \\
& \color{red}{f_{uv} \in \mathbb{R},} & \color{red}{\text{for each } u, v \in V} &
\end{array}
$$

Back to Maximum Flow

Does the LP-relaxation of this ILP always have an integer solution?

$$
\begin{array}{llll}
\text{maximise} & \displaystyle\sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} & & \\
\text{subject to} & f_{uv} \leq c_{uv}, & \text{for each} & u, v \in V \\
& \displaystyle\sum_{v \in V} f_{vu} = \sum_{v \in V} f_{uv}, & \text{for each } u \in V - \{s, t\} & \\
& f_{uv} \geq 0, & \text{for each} & u, v \in V \\
& f_{uv} \in \mathbb{R}, & \text{for each } u, v \in V &
\end{array}
$$

LPs for Max-Flow and Min-Cut

The Max-Flow problem for integer flows can be written as an ILP.

The Min-Cut problem can be written as an ILP (cuts are always integers).

We can write the LP-relaxations of those two ILPs.

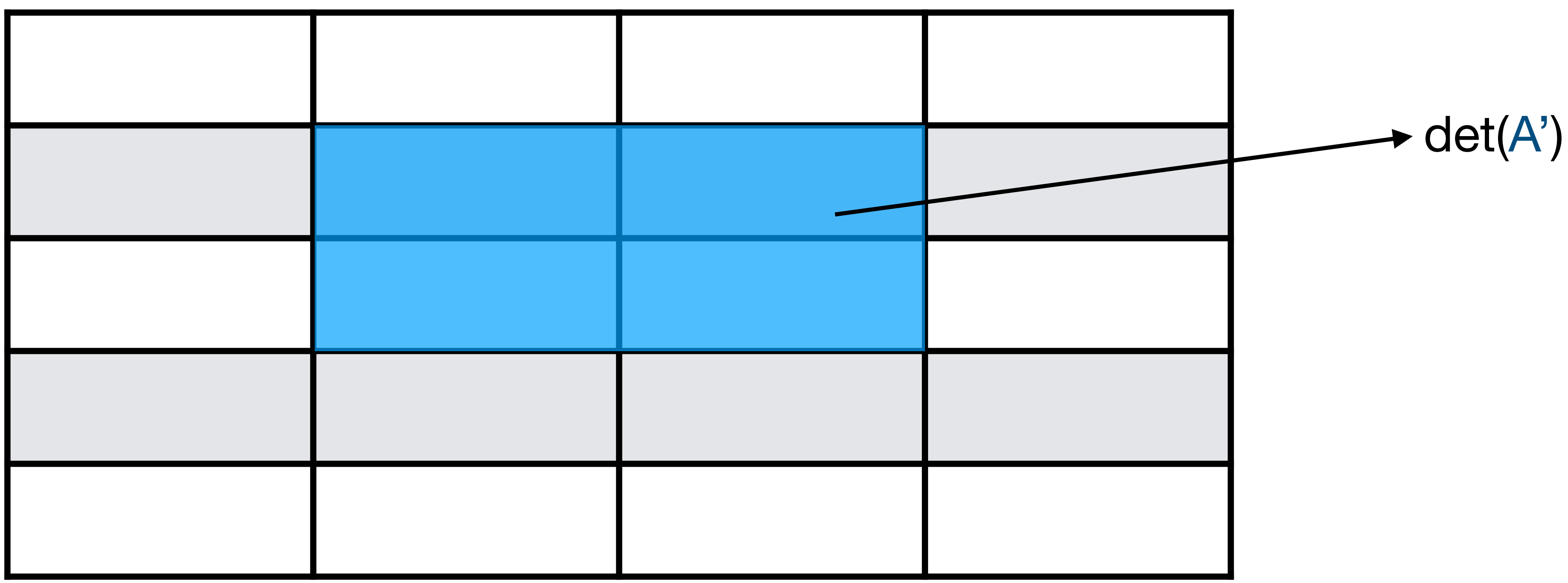
For Max-Flow, it finds the maximum fractional flow.

For Min-Cut, it finds the minimum "fractional" cut.

If we solve those LP-relaxations, we will get *integer solutions*.

Totally Unimodular Matrices

Let A be a real *m* x *n* matrix. Suppose that every square submatrix of A has determinant in {*0*, *+1*, *-1*}. Then A is *totally unimodular*.



Total Unimodularity

If the constraint matrix A is totally unimodular and b is an *integer vector*, then the LP has an *integer solution*.

The Max-Flow and Min-Cut LP-relaxations admit integer solutions because their constraint matrices are totally unimodular.

$$\begin{aligned} \text{maximise} \quad & c^{\mathrm{T}} x \\ \text{subject to} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned}$$

To be more precise

Lemma: Suppose A is a totally unimodular matrix and b is an integer vector. Then every extreme point of

$P = \{x: Ax < b\}$ is integral.

Claim: Suppose A is totally unimodular. Then the matrix $A' = (A\ {-A}\ I\ {-I})^T$ is also totally unimodular.

Corollary: Suppose A is a totally unimodular matrix and b is an integer vector. Then every extreme point of

$P = \{x: Ax = b, 0 \leq x \leq c\}$ is integral.

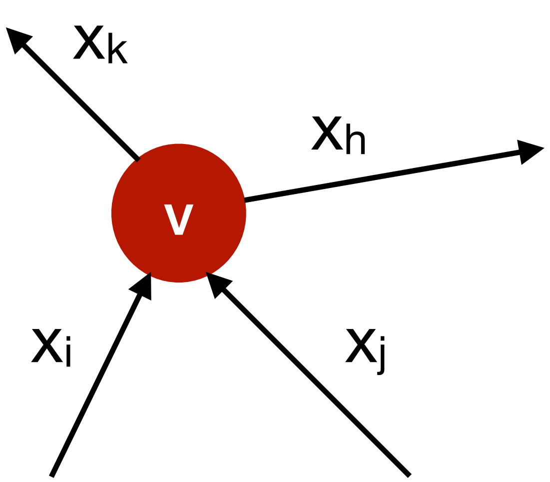
Back to maximum flow

$$\text{maximise} \quad \sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs}$$

$$\text{subject to} \quad \boxed{f_{uv} \le c_{uv}, \quad \text{for each} \quad u, v \in V}$$

$$\boxed{\sum_{v \in V} f_{vu} = \sum_{v \in V} f_{uv}, \quad \text{for each } u \in V - \{s, t\}}$$

$$\boxed{f_{uv} \ge 0, \quad \text{for each} \quad u, v \in V}$$



$x_k + x_h - x_i - x_j = 0$

$0 \le x \le c$

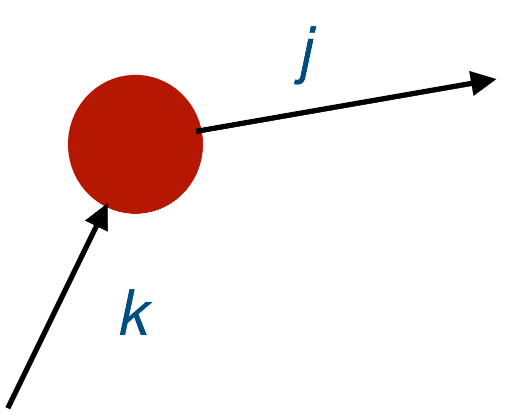
$Ax = 0$

$A_{vi} = A_{vj} = -1$

$A_{vk} = A_{vh} = 1$

Back to maximum flow

- Consider the *incidence matrix* of the flow network (without s and t):
  - $A_{ij} = 1$ if edge j **starts** at node i in G_f.
  - $A_{ij} = -1$ if edge j **ends** at node i in G_f.
  - $A_{ij} = 0$ otherwise.



Nodes/Edges		*j*	*k*
i		1	-1

Back to maximum flow

Consider the *incidence matrix* of the flow network (without s and t):

This is precisely the matrix A of the max flow LP.

It suffices to prove that A is totally unimodular, by Corollary.

Lemma: The incidence matrix of any directed graph is totally unimodular.

Nodes/Edges		*j*	*k*
i		1	-1