Linear Programming II
Week 10

COMP 1201 (Algorithmics)

ECS, University of Southampton

15 May 2020

Previously...

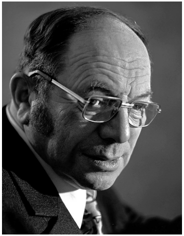
Linear Programming

- Studied by **Leonid Kantorovich** and **Tjalling Koopmans** around 1939.
- A class of **optimisation problems**.
- Optimising **linear functions**, e.g.

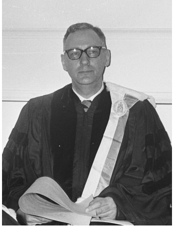
$$3x + y - 2z$$

subject to constraints described by linear functions, e.g.

$$x + y + z \leq 0, \;\; x \geq 0, \;\; y \geq 0, \;\; z \geq 0.$$



L Kantorovich



T C Koopmans

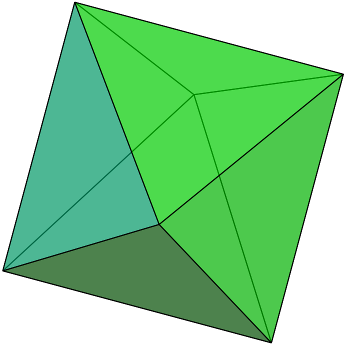
The Simplex Method

Simplex Algorithm

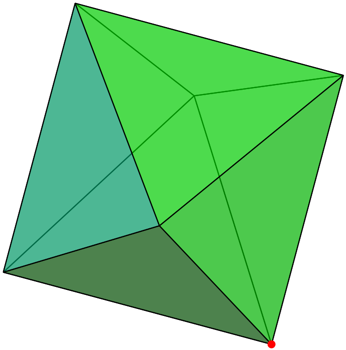
- Invented by **George Dantzig** in 1947.
- The Simplex Algorithm is to this day considered to be the standard method for solving linear programs.
- Remarkable for its practical efficiency.
- Roughly speaking, Simplex is an iterative improvement algorithm that traverses the vertices of the constraint polytope in search of a global optimum.

George B Dantzig
(left, with president Ford)

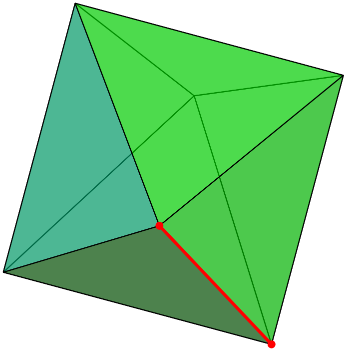
Simplex (intuition)



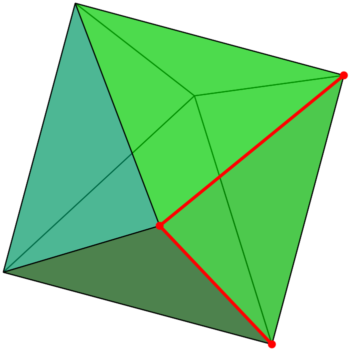
Simplex (intuition)



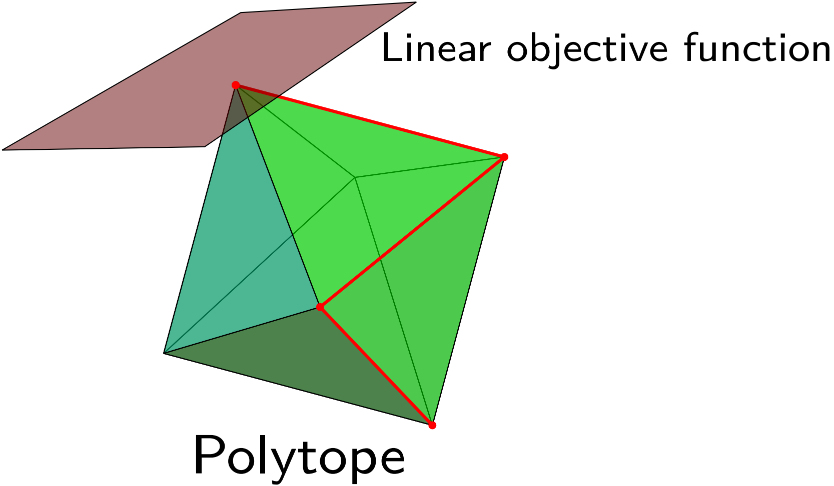
Simplex (intuition)



Simplex (intuition)



Simplex (intuition)



Form of Linear Programs

Linear programs involve a linear objective function

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

which is optimised subject to constraints described by a system of linear inequalities.

To recap, linear programs are problems that can be formulated as follows (using matrix notation):

minimise: $\vec{c} \cdot \vec{x}$,
subject to:
$A_1\vec{x} \leq \vec{b}_1$,
$A_2\vec{x} \geq \vec{b}_2$,
$A_3\vec{x} = \vec{b}_3$,
$\vec{x} \geq \vec{0}$.

Transforming Linear Programs

We can always transform an inequality constraint into an *equality constraint* by adding **slack variables**, e.g. :

$$\vec{a}_1 \cdot \vec{x} \geq 0 \quad \rightarrow \quad \vec{a}_1 \cdot \vec{x} - z_1 = 0,\ z_1 \geq 0.$$
$$\vec{a}_1 \cdot \vec{x} \leq 0 \quad \rightarrow \quad \vec{a}_1 \cdot \vec{x} + z_2 = 0,\ z_2 \geq 0.$$

z_1 (excess) and z_2 (deficit) are known as slack variables.

A linear program featuring only equality constraints $A\vec{x} = \vec{b}$ is said to be in *normal form*.

(N.B. the non-negativity constraints $\vec{x} \geq \vec{0}$ still apply).

Transforming Linear Programs

After eliminating inequality constraints by introducing slack variables, we arrive at a system of constraints

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

of larger dimension (in a sense we are embedding our n-dimensional feasible set polytope in a larger-dimensional space, however the equality constraints restrict solutions to a lower-dimensional sub-space).

We expect there to be more variables than constraints, so we expect the system to be underdetermined. This is usually the case.

Solutions are built from a set of **basic variables**. The other variables are **non-basic** and are typically set to zero.

The Simplex Method

Let us consider the following Linear Program:

$$\begin{aligned} \textbf{maximise :} \quad & 6x_1 + 7x_2 + 9x_3, \\ \textbf{sumbject to:} \quad & 2x_1 + x_2 + 4x_3 \leq 100, \\ & x_1 + x_2 + x_3 \leq 50, \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

The Simplex Method

Let us consider the following Linear Program:

$$\begin{aligned} \textbf{maximise :} \quad & 6x_1 + 7x_2 + 9x_3, \\ \textbf{sumbject to:} \quad & 2x_1 + x_2 + 4x_3 \leq 100, \\ & x_1 + x_2 + x_3 \leq 50, \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

This needs to be transformed into an appropriate form in order to apply the Simplex Method.

The Simplex Method

Let us consider the following Linear Program:

$$\begin{aligned} \textbf{maximise :} \quad & 6x_1 + 7x_2 + 9x_3, \\ \textbf{sumbject to:} \quad & 2x_1 + x_2 + 4x_3 \leq 100, \\ & x_1 + x_2 + x_3 \leq 50, \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

This needs to be transformed into an appropriate form in order to apply the Simplex Method.

We need to eliminate two inequality constraints by introducing slack variables s_1 and s_2.

The Simplex Method

Let us consider the following Linear Program:

$$
\begin{aligned}
\textbf{maximise :} \quad & 6x_1 + 7x_2 + 9x_3, \\
\textbf{sumbject to:} \quad & 2x_1 + x_2 + 4x_3 + s_1 = 100, \\
& x_1 + x_2 + x_3 + s_2 = 50, \\
& x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, s_1 \geq 0, s_2 \geq 0.
\end{aligned}
$$

This needs to be transformed into an appropriate form in order to apply the Simplex Method.

We need to eliminate two inequality constraints by introducing slack variables s_1 and s_2.

The Simplex Method

Let us consider the following Linear Program:

$$\begin{aligned} \textbf{maximise :} \quad & 6x_1 + 7x_2 + 9x_3, \\ \textbf{sumbject to:} \quad & 2x_1 + x_2 + 4x_3 + s_1 = 100, \\ & x_1 + x_2 + x_3 + s_2 = 50, \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, s_1 \geq 0, s_2 \geq 0. \end{aligned}$$

This needs to be transformed into an appropriate form in order to apply the Simplex Method.

We need to eliminate two inequality constraints by introducing slack variables s_1 and s_2.

We should also introduce a new variable f for the objective function, and rewrite it as an equation (in non-basic variables).

The Simplex Method

Let us consider the following Linear Program:

$$
\begin{aligned}
\textbf{maximise :}\quad & f, \\
\textbf{sumbject to:}\quad & 2x_1 + x_2 + 4x_3 + s_1 = 100, \\
& x_1 + x_2 + x_3 + s_2 = 50, \\
& -6x_1 - 7x_2 - 9x_3 + f = 0, \\
& x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, s_1 \geq 0, s_2 \geq 0.
\end{aligned}
$$

This needs to be transformed into an appropriate form in order to apply the Simplex Method.

We need to eliminate two inequality constraints by introducing slack variables s_1 and s_2.

We should also introduce a new variable f for the objective function, and rewrite it as an equation (in non-basic variables).

The Simplex Method (Example)

$$
\begin{aligned}
\textbf{maximise :}\quad & f, \\
\textbf{sumbject to:}\quad & 2x_1 + x_2 + 4x_3 + s_1 = 100, \\
& x_1 + x_2 + x_3 + s_2 = 50, \\
& -6x_1 - 7x_2 - 9x_3 + f = 0, \\
& x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, s_1 \geq 0, s_2 \geq 0.
\end{aligned}
$$

- Step 1: Write the transformed Linear Program down in tabular form.

	x_1	x_2	x_3	s_1	s_2	f	
s_1							
s_2							
f							

The Simplex Method (Example)

$$
\begin{aligned}
\textbf{maximise :}\quad & f, \\
\textbf{sumbject to:}\quad & 2x_1 + x_2 + 4x_3 + s_1 = 100, \\
& x_1 + x_2 + x_3 + s_2 = 50, \\
& -6x_1 - 7x_2 - 9x_3 + f = 0, \\
& x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, s_1 \geq 0, s_2 \geq 0.
\end{aligned}
$$

- Step 1: Write the transformed Linear Program down in tabular form.

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

In this example they immediately provide us with an **initial feasible solution** (i.e. a vertex of the polytope) which is required for Simplex.

In general, efficiently finding an initial feasible solution can be tricky.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

In this example they immediately provide us with an **initial feasible solution** (i.e. a vertex of the polytope) which is required for Simplex.

In general, efficiently finding an initial feasible solution can be tricky.

- Step 2: Scan the last row in the table and select the column with the smallest negative entry as the *pivot column*.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 2: Scan the last row in the table and select the column with the smallest negative entry as the *pivot column*.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 2: Scan the last row in the table and select the column with the smallest negative entry as the *pivot column*.
- For each entry above the objective function coefficient in the pivot column, divide the corresponding constant term in the last column by that entry.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 2: Scan the last row in the table and select the column with the smallest negative entry as the *pivot column*.
- For each entry above the objective function coefficient in the pivot column, divide the corresponding constant term in the last column by that entry. I.e. $100/4 = 25$

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 2: Scan the last row in the table and select the column with the smallest negative entry as the *pivot column*.
- For each entry above the objective function coefficient in the pivot column, divide the corresponding constant term in the last column by that entry. I.e. $100/4 = 25$, $50/1 = 50$.
- Select the entry from the pivot column where the resulting value was smallest as the *pivot element*, and divide the corresponding row to make the pivot 1.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	2	1	4	1	0	0	100
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 2: Scan the last row in the table and select the column with the smallest negative entry as the *pivot column*.
- For each entry above the objective function coefficient in the pivot column, divide the corresponding constant term in the last column by that entry. I.e. $100/4 = 25$, $50/1 = 50$.
- Select the entry from the pivot column where the resulting value was smallest as the *pivot element*, and divide the corresponding row to make the pivot 1.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	$1/2$	$1/4$	1	$1/4$	0	0	25
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 2: Scan the last row in the table and select the column with the smallest negative entry as the *pivot column*.
- For each entry above the objective function coefficient in the pivot column, divide the corresponding constant term in the last column by that entry. I.e. $100/4 = 25$, $50/1 = 50$.
- Select the entry from the pivot column where the resulting value was smallest as the *pivot element*, and divide the corresponding row to make the pivot 1.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	$1/2$	$1/4$	1	$1/4$	0	0	25
s_2	1	1	1	0	1	0	50
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 3: Perform row reductions to make all other entries in the pivot column (except the pivot element) equal to 0.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	$1/2$	$1/4$	1	$1/4$	0	0	25
s_2	$1/2$	$3/4$	0	$-1/4$	1	0	25
f	-6	-7	-9	0	0	1	0

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 3: Perform row reductions to make all other entries in the pivot column (except the pivot element) equal to 0.
- $R_2 := R_2 - R_1$

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
s_1	$1/2$	$1/4$	1	$1/4$	0	0	25
s_2	$1/2$	$3/4$	0	$-1/4$	1	0	25
f	$-3/2$	$-19/4$	0	$9/4$	0	1	225

The variables s_1 and s_2 in this particular table are the **basic variables**; all the other variables are **non-basic**.

- Step 3: Perform row reductions to make all other entries in the pivot column (except the pivot element) equal to 0.
- $R_3 := R_3 + 9R_1$

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	$1/2$	$1/4$	1	$1/4$	0	0	25
s_2	$1/2$	$3/4$	0	$-1/4$	1	0	25
f	$-3/2$	$-19/4$	0	$9/4$	0	1	225

- Repeat the process until there are no negative coefficients left in the bottom row.
- (Note that x_3 has now replaced s_1 as a **basic variable**.)

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	1	$1/2$	2	$1/2$	0	0	50
s_2	$1/2$	$3/4$	0	$-1/4$	1	0	25
f	$-3/2$	$-19/4$	0	$9/4$	0	1	225

- Repeat the process until there are no negative coefficients left in the bottom row.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	1	$1/2$	2	$1/2$	0	0	50
s_2	$2/3$	1	0	$-1/3$	$4/3$	0	$100/3$
f	$-3/2$	$-19/4$	0	$9/4$	0	1	225

- Repeat the process until there are no negative coefficients left in the bottom row.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	$2/3$	0	2	$2/3$	$-2/3$	0	$100/3$
s_2	$2/3$	1	0	$-1/3$	$4/3$	0	$100/3$
f	$-3/2$	$-19/4$	0	$9/4$	0	1	225

- Repeat the process until there are no negative coefficients left in the bottom row.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	$2/3$	0	2	$2/3$	$-2/3$	0	$100/3$
s_2	$2/3$	1	0	$-1/3$	$4/3$	0	$100/3$
f	$5/3$	0	0	$2/3$	$19/3$	1	$1150/3$

- Repeat the process until there are no negative coefficients left in the bottom row.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	$1/3$	0	1	$1/3$	$-1/3$	0	$50/3$
x_2	$2/3$	1	0	$-1/3$	$4/3$	0	$100/3$
f	$5/3$	0	0	$2/3$	$19/3$	1	$1150/3$

- Make sure the columns with basic variables are in the correct form.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	$1/3$	0	1	$1/3$	$-1/3$	0	$50/3$
x_2	$2/3$	1	0	$-1/3$	$4/3$	0	$100/3$
f	$5/3$	0	0	$2/3$	$19/3$	1	$1150/3$

- We are now done!
- x_3 and x_2 are now the basic variables.

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	$1/3$	0	1	$1/3$	$-1/3$	0	$50/3$
x_2	$2/3$	1	0	$-1/3$	$4/3$	0	$100/3$
f	$5/3$	0	0	$2/3$	$19/3$	1	$1150/3$

- We are now done!
- x_3 and x_2 are now the basic variables.
- The solution is given by $x_3 = 50/3$, $x_2 = 100/3$ (basic variables), $x_1 = 0, s_1 = 0, s_2 = 0$ (non-basic variables, which are set to zero).

The Simplex Method (Example)

	x_1	x_2	x_3	s_1	s_2	f	
x_3	$1/3$	0	1	$1/3$	$-1/3$	0	$50/3$
x_2	$2/3$	1	0	$-1/3$	$4/3$	0	$100/3$
f	$5/3$	0	0	$2/3$	$19/3$	1	$1150/3$

- We are now done!
- x_3 and x_2 are now the basic variables.
- The solution is given by $x_3 = 50/3$, $x_2 = 100/3$ (basic variables), $x_1 = 0, s_1 = 0, s_2 = 0$ (non-basic variables, which are set to zero).
- The maximum of our objective function is $1150/3$.

High Performance Solvers

- The tableau method is rather simplistic and isn't the best option for solving large-scale linear programs.
- Updates in Simplex can be viewed as solving a set of linear equations which is facilitated by performing LU-decompositions.
- The constraints are often sparse and good solvers try to take advantage of the sparsity.
- Top end Simplex implementations are rather complex.
- Simplex is not the only algorithm for solving linear programs.

Time Complexity of Simplex

- It turns out that **typically** Simplex runs in $O(n^3)$ time in practice.
- The main question is how many "hops" are necessary.
- However, it is possible to cook up problems where there is a "long path" from the initial solution to the optimum, which is exponentially large.
- Thus, the theoretical worst-case time complexity of the Simplex algorithm is **exponential** (although in practice this almost never happens).

Interior Point Methods

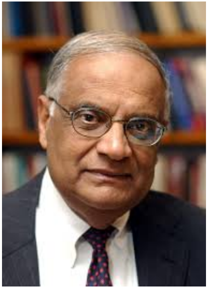
Ellipsoid Method

- Pioneered by **Leonid Khachiyan** in 1979.
- The first **polynomial time** algorithm for solving Linear Programs.
- Caused a big stir when it was discovered.
- However, was utterly impractical for solving any real world problems.
- Could not compete with Simplex.

Interior Point Methods

Karmarkar's Algorithm

- Invented by **Narendra Karmarkar**, 1984.
- The first **practical** alternative to Simplex.
- Runs in **polynomial time**.
- Instead of hopping from vertex to vertex like Simplex, interior point algorithms traverse the *interior* of the feasible region.
- Changed the landscape of LP profoundly.
- Led to renewed interest and research in interior point methods for solving LP.



Narendra Karmarkar

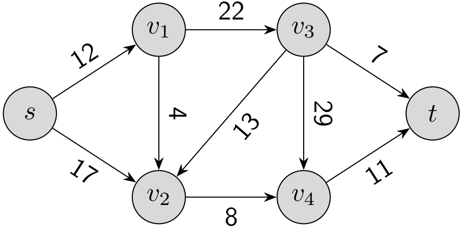
Interior Point Methods

- Today interior point algorithms compete with (variations of) Simplex on very large problems.
- A number of good Linear Programming solver implementations are making use of these algorithms.
- The two families of methods are complementary and it can be difficult to determine in which cases one is better than the other.
- In practice, it helps to have access to both.

When is LP Appropriate? (Maximum Flow)

- In maximum flow we consider a directed graph representing a network of pipes.
- We choose one vertex as the **source** and another vertex as the **sink**.
- Each edge in the graph has a **flow capacity** that cannot be exceeded.
- The problem is to maximise the flow between the source and sink.
- This can be used to model the flow of a fluid, parts in an assembly line, current in an electrical circuit, or packets through a communications network.

Maximum Flow



This is such a classic problem with a distinctive structure that we can solve it more efficiently using other algorithms developed specifically for this problem. The classic algorithm is the

Ford-Fulkerson method (see e.g. CLRS), which is $O(|E| \times f_{\max})$, where $f_{\max}$ is the maximum flow.

Linear Assignment

- We are given a set of n agents A, and a set of n tasks T.
- Each agent has a cost associated with performing a task $c(a, t)$.
- We want to assign an agent to one task so as to minimise the total cost.
- Consider a taxi firm with taxis at 5 different locations and 5 requests to fulfil. The cost is the distance to the client. Which taxi should go to which client?

Linear Assignment

- The linear assignment problem can be set as a linear programming problem:
- The objective function to minimise is:

$$\sum_{a \in A,\ t \in T} c(a,t) x_{a,t}$$

- The constraints are:

$$\forall\ a \in A.\ \sum_{t \in T} x_{a,t} = 1,$$

$$\forall\ t \in T.\ \sum_{a \in A} x_{a,t} = 1,$$

$$\forall\ (a,t) \in A \times T.\ x_{a,t} \geq 0.$$

Linear Assignment

- Although linear assignment can be solved using a generic LP solver, this is not the most efficient solution.
- A more efficient solution is the Hungarian Algorithm.
- This is rather complex.
- The worst case time complexity is $O(n^3)$, although it frequently takes $O(n^2)$.

Quadratic Programming

- An optimisation problem with linear constraints and a **quadratic** objective function is said to be a **quadratic programming problem**.
- Such a problem can likewise be solved in polynomial time.
- Many of the ideas are similar to LP.
- Important applications in science and engineering.

Lessons

- Linear programming is a classic problem.
- A **huge number** of problems are solvable in polynomial time because they can be formulated as linear programs.
- Linear programs occur sufficiently often that they are hugely important in practice.
- They are generally **not easy to solve**; however, the basic Simplex algorithm is not massively complex.
- Some important problems can be solved using linear programming, but possess special structure for which there are **better algorithms**.

Further Reading:

1. **Jiří Matoušek, Bernd Gärtner** “Understanding and Using Linear Programming”
https://link.springer.com/book/10.1007/978-3-540-30717-4

Optional (open problems):

- **Stephen Smale** “Mathematical Problems for the Next Century”, *Problem 9: The Linear Programming Problem*.
https://link.springer.com/content/pdf/10.1007/BF03025291.pdf
(Problems 3, 5, 17 and 18 are also computer science problems; 17 has been solved, the rest are still open.)

Acknowledgements: Partly based on earlier COMP 1201 slides by Dr Adam Prügel-Bennett, University of Southampton.