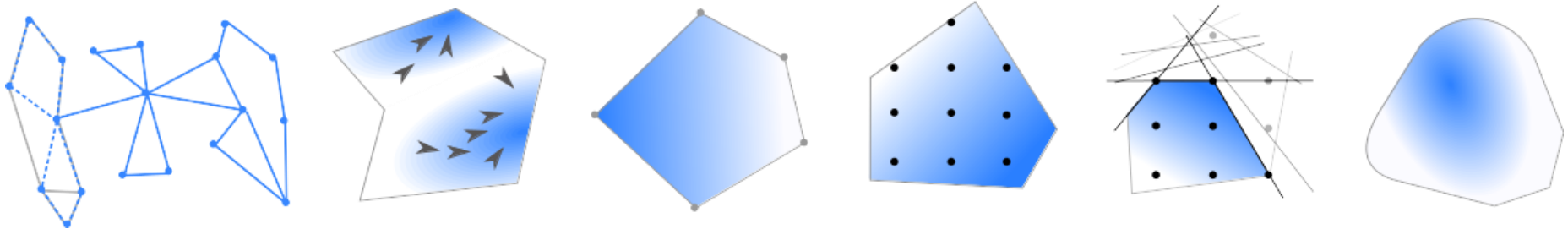# Optimisation

COMP4691 / 8691

Course Overview

Lecturer/Convener: Felipe Trevizan

Tutors

Jiawen Wang

Jerry Zhang

Course Outline

Main learning outcome of the course:

- Modelling and implementation of optimisation problems and algorithms

Main topics:

- Linear Programming
- Mixed-Integer Linear Programming
- Decomposition
- Convex Optimisation
- Local Search
- Metaheuristics
- Advanced Topics: Stochastic Opt., Multi-Objective Opt., Guest Lectures

COMP4691/8691 Website

All the information about the course can be found in this website:
https://comp.anu.edu.au/courses/comp4691/

Activities

- Lectures
  - Tuesdays noon–2pm Bldg 48A Rm 1.23
  - Fridays 10–noon **Bldg 48 Lecture Theatre**
- Labs and Drop-ins
  - **Start next week**
  - Wednesdays 4-6 pm N111 CSIT
  - Thursdays 10-noon N109 CSIT
  - Thursdays 3:**30**-5:**30**pm N115/116 CSIT

Lectures

COMP4691/8691   Activities   Deliverables   **Resources**

Books and links

- We provide all the required course material

- If you require additional resources, there are a number of textbooks that we recommend in the resources section

Week	Dates	Lecture A	Lecture B	Labs A/B
1	22/7 - 26/7	01-course-overview -and-introduction	02-LP-modelling	-
2	29/7 - 2/8	03-LP-feasibility -and-optimality	04-LP-simplex	Lab LP: Q1,11,12
3	5/8 - 9/9	05-LP-approximations -and-duality	06-MIP-relaxation -and-modelling	Drop-in
4	12/8 - 16/8	07-MIP-branch	08-MIP-cutting	MIP: Q2,4,7
5	19/8 - 23/8	09-decomp-column-gen	10-decomp-benders	Drop-in
6	26/8 - 30/8	11-cvx-convexity	**NO-LECTURE**	
7	16/9 - 20/9	12-cvx-optimisation -and-lagrangian	13-cvx-interior	Lab Decomposition
8	23/9 - 27/9	14-construction	15-local-search	CVX: Q2,5,8
9	30/9 - 4/10	16-metaheuristics-1	17-metaheuristics-2	
10	7/10 - 11/10	18-stochastic opt	19-multi-objective opt	Lab Metaheuristics
11	14/10 - 18/10	20-network-flow	21-path-planning	Drop-in
12	21/10 - 25/10	Guest Lecture A TBD	Guest Lecture B TBD	

Computer labs/drop-ins

5 Labs (weeks 2, 4, 7, 8, 10):

- 30 minutes drop-in at the start
- 90 minutes working in a **set of problems** to practice your skills, test your knowledge, and prepare you for the assignments and exam.
- The GitLab link to **problem sets** is available in the website.

3 Drop-ins only (weeks 3, 5, 11):

- 2 hours of drop-in
- Use it to ask questions about the lab's problem sets, deliverables, and lectures

Deliverables

Assessment marks

Assessment	COMP4691	COMP8691
Final Exam	50% (40% hurdle)	50% (40% hurdle)
Assignment 1	10%	10%
Assignment 2	10%	10%
5 Quizzes	30%	20%
Seminar	–	10%

Assignments

General instructions

- Assignment 1: LPs and MILPs due: **Friday 30-08-2024 18:00**
- Assignment 2: Meta-heuristics, due: **Week 12**

Quizzes:

- In-person at the end of the lectures
- 30-45 minutes

Timeline

Week	Dates	Due	Released	Quiz
1	22/7 - 26/7		Seminar (COMP8691-only)	
2	29/7 - 2/8			
3	5/8 - 9/9			Quiz-1
4	12/8 - 16/8		Assignment 1	
5	19/8 - 23/8			Quiz-2
6	26/8 - 30/8	Assignment 1		
7	16/9 - 20/9	Seminar: Group and Topic Selection		
8	23/9 - 27/9			Quiz-3
9	30/9 - 4/10		Assignment 2	
10	7/10 - 11/10			Quiz-4
11	14/10 - 18/10			
12	21/10 - 25/10	Assignment 2 Seminar: Video delivery		Quiz-5

Deliverables – 8691 Only

Seminar Overview (COMP8691 ONLY)

Students will work individually or in groups of up to 3 to research an optimisation topic that is not covered in the course material. Topics may include:

- A particularly interesting application of optimisation
- An optimisation algorithm or solving technique

The groups first need to propose a topic that will then be approved (or not) by the course lecturers. Be sure to do this as soon as you can so there is enough time to find another topic if your first choice is taken or inappropriate.

Once the topic is approved, the group will then further research the topic and put together a 15 minute video presentation on it.

- Week 1: Seminar project released
- Week 7: Group and topic selection deadline: **20-09-2024 18:00**
- Week 12: Video delivery deadline: **25-10-2024 18:00**

Resources

CATEGORIES

- General
- Lectures
- Labs
- Async-lab
- Assignment-1
- Assignment-2
- Assignment-3
- Assignment-4
- Exam
- Seminar-groups
- Seminar-topics

RESOURCES

FAQ

Books and Links

Software

RELATED SITES

Ed Discussions

GitLab

Wattle Gradebook

🏠 » Resources

FAQ

Some questions which come up, well… frequently

» read more

Books and Links

Books, links and other resources

» read more

Software

Software setup and usage

» read more

Assignments and lab problems will be available in the GitLab page of the course

Communication

Admin questions: ← <u>the whole COMP4691 team</u>
- Use Ed Discussions
- make it a private message if needed

Admin questions regarding private matters: ← <u>just me</u>
- email the convener directly

Class Representative

The role of Student Representatives is to provide ongoing constructive feedback on behalf of the student cohort to Course Conveners and to Associate Directors (Education) for continuous improvements to the course.

- Act as the official **liaison between your peers and convener.**
- Be available and proactive in **gathering feedback from your classmates**.
- Attend regular meetings, and **provide reports on course feedback** to your course convener
- Close the feedback loop by **reporting back to the class the outcomes of your meetings**.

Note: Class representatives will need to be comfortable with their contact details being made available via Wattle to all students in the class.
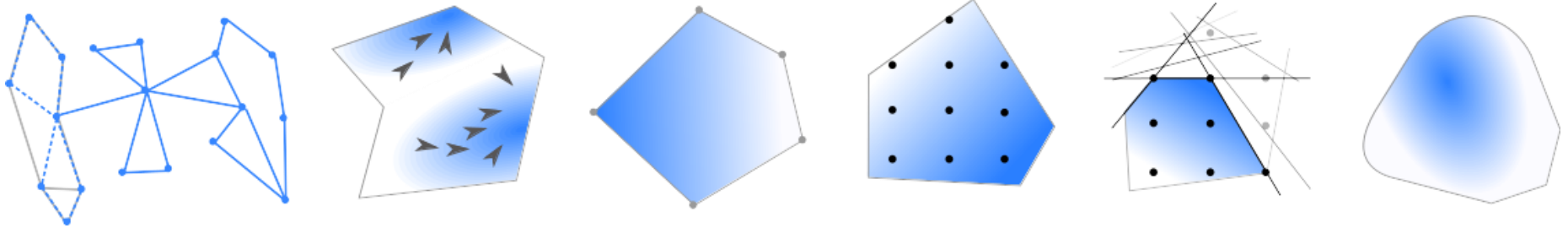
For more info regarding roles and responsibilities, contact: sa.cecs@anu.edu.au

Please nominate yourself by sending a private message in ED

Optimisation

COMP4691 / 8691

Introduction to optimisation

Agenda

- What is optimisation?
- Examples of real-world optimisation problems
- How do we solve optimisation problems?
- Problem Formulation
- Methods to solve optimisation problems

Optimisation

… is the selection of a best element, with regard to some criterion, from some set of available alternatives …

Wikipedia

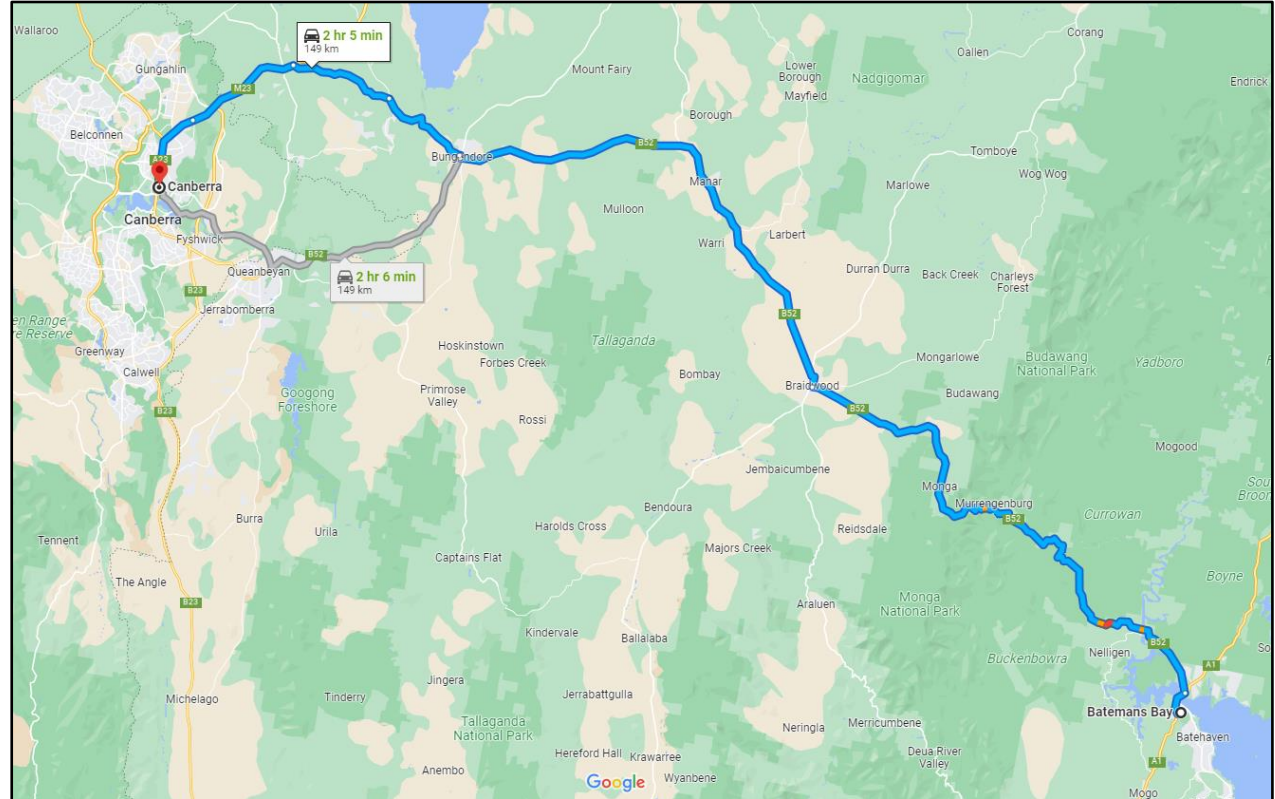… finding the best solution out of a very large set of possible solutions.

Google

The art of making choices, when good choices lead to good solutions.

Phil Kilby

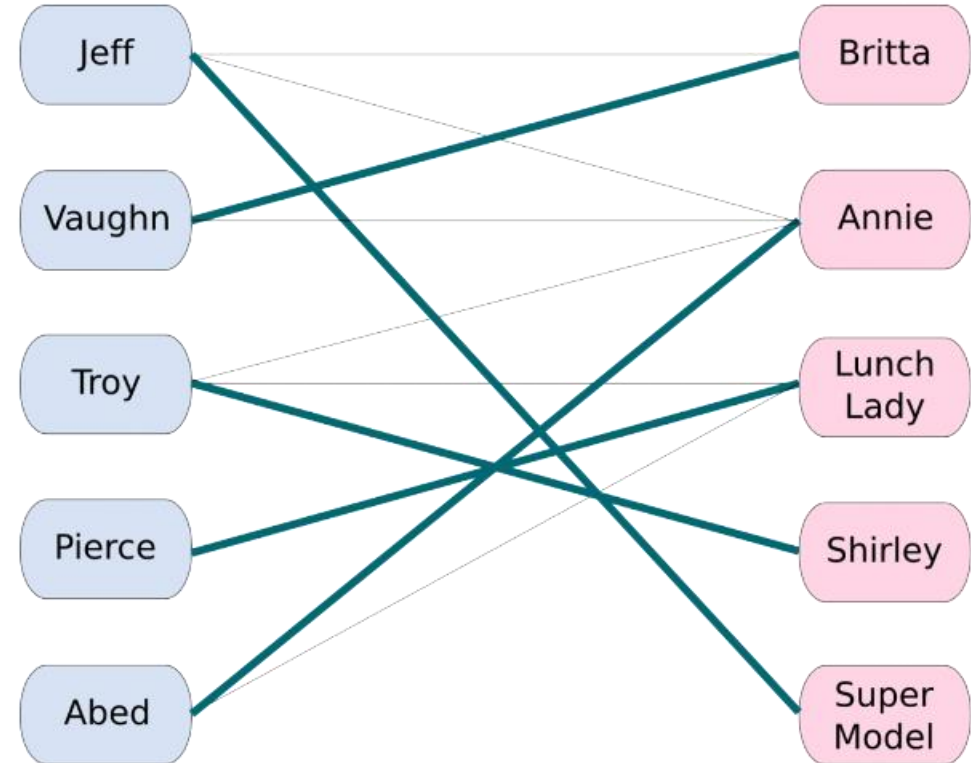# Examples of optimisation problems

Shortest path problem

- Find shortest path between two locations

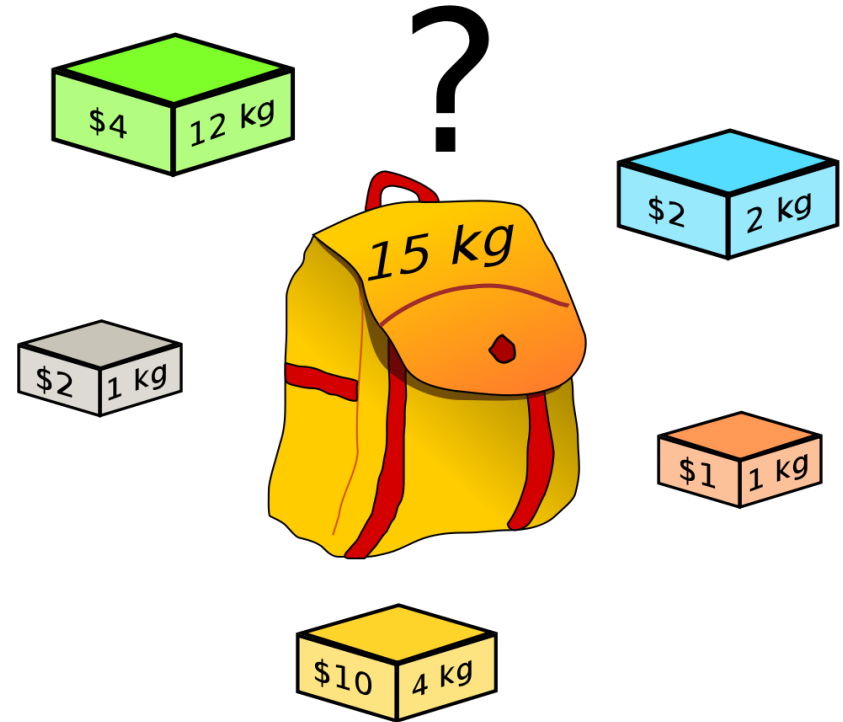# Examples of optimisation problems

Matching problems

- Find the pairing with minimum cost (or maximal satisfaction)

Examples of optimisation problems

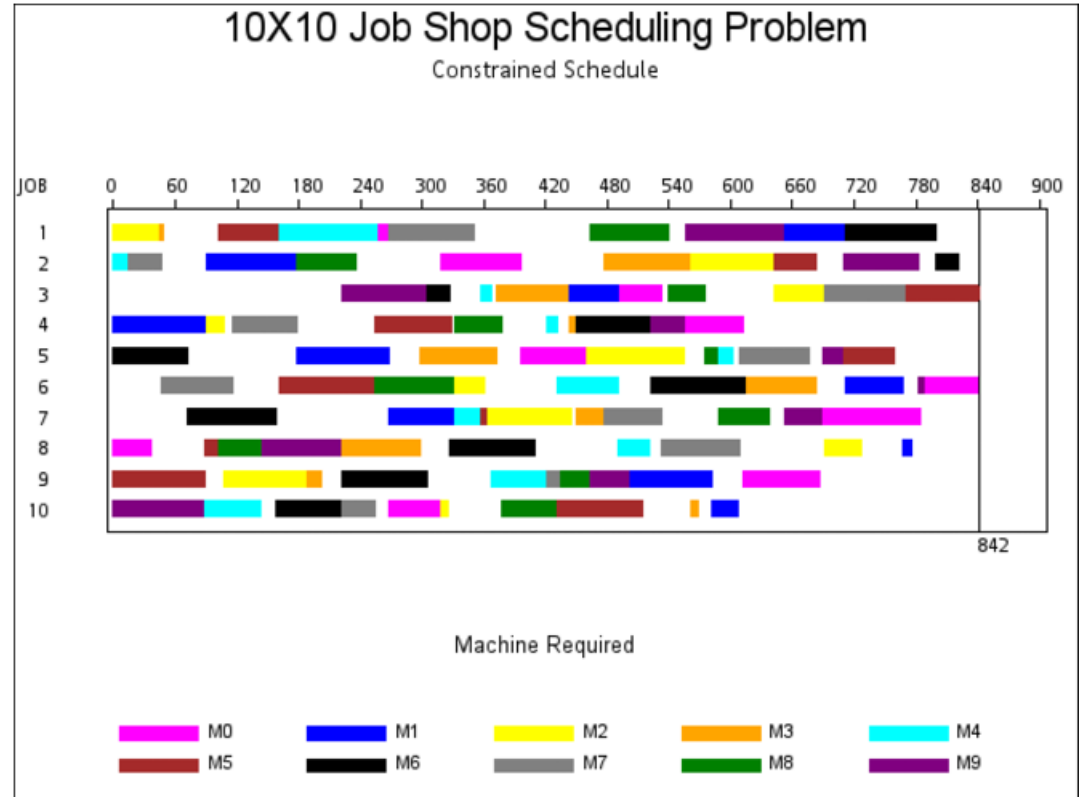## Knapsack / Bin Packing problem

- Choose items to give the maximum value within a given capacity

Examples of optimisation problems
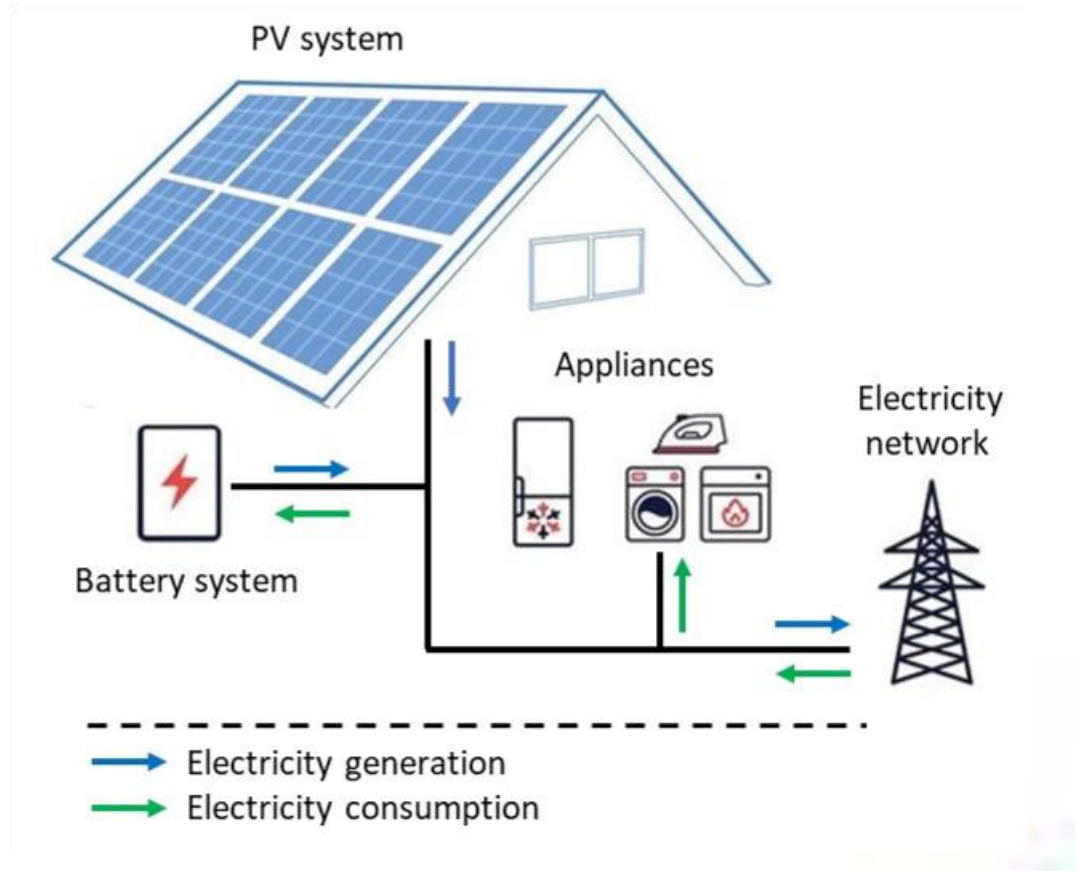
Job-shop scheduling problem

- Schedule jobs on machines to minimise total time

Examples of optimisation problems

Smart home scheduling problem

- Scheduling the operation of a smart home with a PV-battery system

PV system

Appliances

Electricity network

Battery system

→ Electricity generation
→ Electricity consumption

Examples of optimisation problems

Electricity market dispatch problem

- Dispatch the operation of generators to meet demand by maximising social-welfare (or minimising costs)
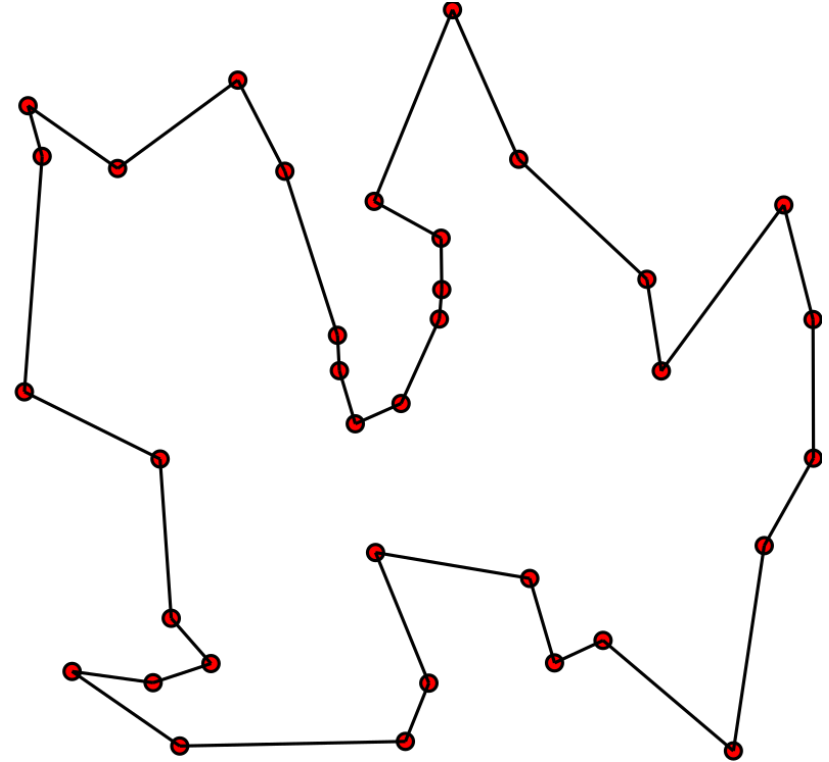

Source: AEMO
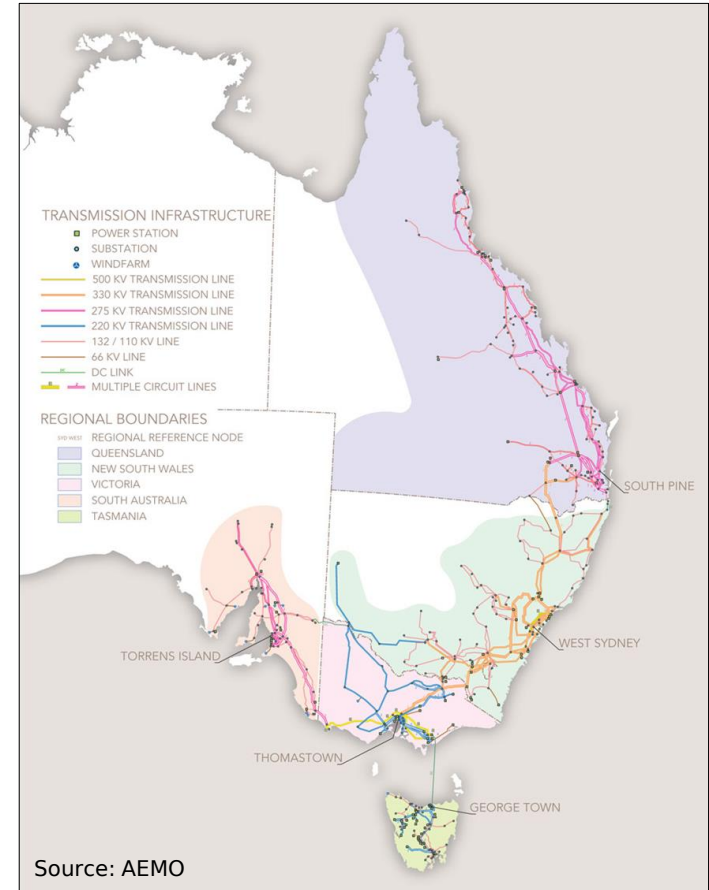
Examples of optimisation problems

Travelling salesman problem

- The travelling salesman problem asks the following question:

  "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"

Examples of optimisation problems

Vehicle routing problem

- Given a set of customers, and a fleet of vehicles to make deliveries, find a set of routes that services all customers at minimum cost
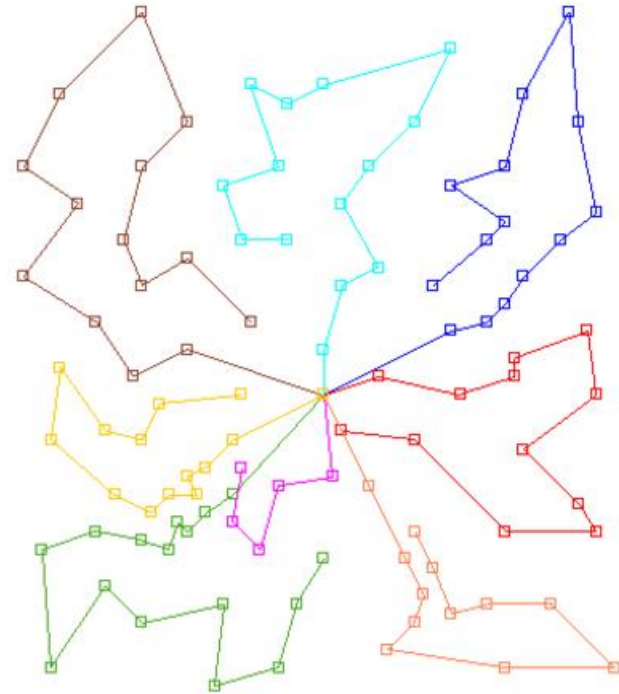
What optimisation problems did you learn on AI (COMP3620)?

How do we solve these optimisation problems?

How do we solve these optimisation problems?

Typically, we need to follow these steps:

1. Formulate the optimisation problem

2. Select the method to solve the problem

Problem Formulation

Definition

- Translating the statement of a problem into a mathematical formulation

Key elements

- Decision variables

- Parameters

- Constraints

- Objective(s)

Decision variables

- Variables that we have control over (that affect our solution)

- Example of decision variables in the smart home scheduling problem:
  - Power generation of the PV system
  - Discharging and charging power of the battery system

- Example of decision variables for the job-shop scheduling problem:
  - Variables to select which machines operate
  - Starting time of the machines

Parameters

- Elements we have no control over (but do effect the outcome)
- **They are constant**

- Example of parameters in the smart home scheduling problem:
  - Efficiency of the battery
  - Capacity of the battery

- Example of parameters for the job-shop scheduling problem:
  - Number of machines
 - Execution time of each job

Constraints

- Define:
  - the limits of the decision variables
  - relationships between variables
  - relationships between variables and parameters

- Example of constraints in the smart home scheduling problem:
  - Maximum charging and discharging power of the battery
  - Exchanged power with the network

- Example of constraints for the job-shop scheduling problem:
  - Assign jobs to machines
  - Coordinate the operation of machines: machine M0 starts before than M3

Objective(s)

- What makes a solution "good"?

- What characteristics are we seeking in a solution?
  - Smart home problem: minimise the electricity cost of the house
  - Scheduling problem: minimise the maximum completion time over all jobs
  (makespan)

- Sense is minimisation (e.g., of cost) or maximisation (e.g., of profit)

- Multiple objectives:
  - Smart home problem: minimise electricity cost and environmental impacts

Methods to solve optimisation problems
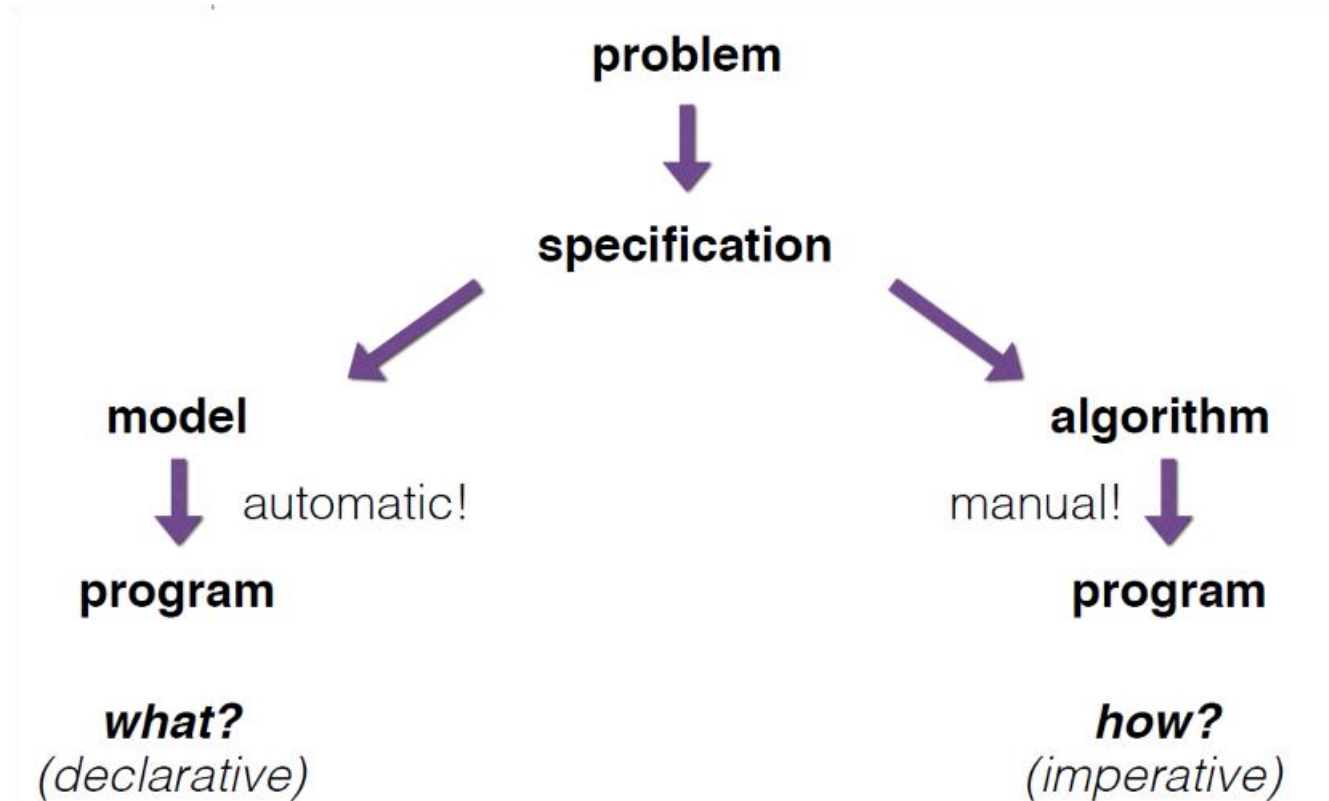
Model-based methods

Examples of model-based methods:

- LP (Linear programming) – we will start this next class

- MILP (Mixed-integer linear programming) – we will study this later

- SOCP (Second-order cone programming)

- QP (Quadratic programming)

- SDP (Semidefinite programming)

These methods can handle a class of problems

The models are inputs to the solvers

Model-based methods

Advantages

- Formulate the problem once
- Separates problem formulation from input data
- Automatic solution techniques (e.g., solvers)
- We can add / remove constraints easily
- Guarantees of optimal solution in convex problems

Disadvantages

- It can be slower than algorithm-based methods
- Less control over the solution method

Algorithm-based methods

Algorithm-based methods

- Method is specifically designed for a particular problem (or class of problems)

- Often rely on "rules of thumb" to guide the method to a good solution

- Examples:

  - Heuristics (often problem-dependent) – we will study this later

  - Metaheuristics (problem-independent techniques that can be applied to a class of problems) – we will study this later

Algorithm-based methods

Advantages

- It can usually solve the problem much faster
- Tailor solution method

Disadvantages

- Tailored to a specific problem statement
- New constraint → rewrite algorithm?
- Time-consuming to develop
- Trial-and-error – intuition does not always work

Summary

We've looked at

- Some optimisation problems

- Formulating a problem

- Model-based and algorithm-based methods

We will look at most of these aspects in more detail over the coming weeks.

We will start with linear problems.