1 Session 1: Introduction

- Learn about the origins and applications of Operations Research
- Understand system modelling principles
- Understand algorithm efficiency and problem complexity
- Contrast between the optimality and practicality
- Learn about software for operations Research
- Introduction to the Python/Colab environment

1.1 Operation research

The use of quantitative methods to assist analysis and decision-makers in designing, analyzing and improving the performance or operation of systems.

...or...

Applied math field, where mathematical tools and operators aren't used to investigate mathematics further but rather to analyse and solve problems within the OR domain by designing innovative solution approaches.

Link to a big picture view of OR

Some examples:[1]

- Dynamic programming: How should one design an optimal policy for buying stock options? How should a project manager distribute collective effort to fulfil the deliverables and minimise the time span of the project?
- Integer and linear programming: How should radiotherapists design a treatment plan for their cancer patients? How should a manufacturer price its resources? What is the shortest tour through the 50 states in the United States? How does a computer solve a Sudoku puzzle?
- Markov chains: How should an internet search engine rank trillions of results when a search takes place? In Snakes and Ladder, how do the expected number of moves to the finish line differ depending on where you are on the board?
- Queueing theory: How should a high-performance computing facility assign jobs to its servers? How many tellers should you hire so that 90% of the bank customers wait at most 5 minutes before being served?
- Game theory: Faced with stiff competition in the marketplace, how should you pick your strategy or strategies? What is the payoff of a strategy?

[1] https://www.lse.ac.uk/study-at-lse/summer-schools/summer-school/courses/research-methods/me205

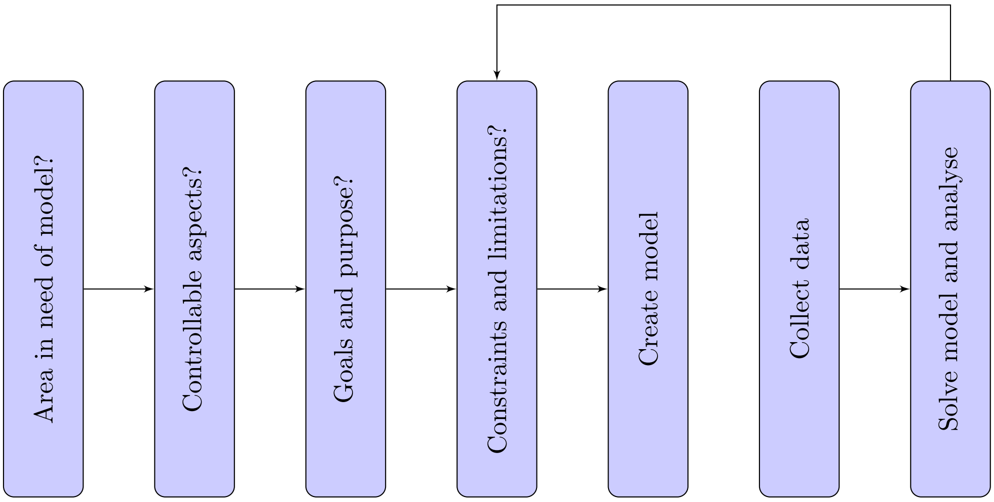
https://www.britannica.com/topic/operations-research/Essential-characteristics
https://money.usnews.com/careers/best-jobs/operations-research-analyst
https://towardsdatascience.com/some-thoughts-on-synergies-between-operations-resea
https://medium.com/berk-orbay/an-ode-to-operations-research-and-the-future-757445
https://feasible.substack.com/p/23-part-ii-the-or-revolution-3-ai
https://arxiv.org/html/2401.03244v1
https://www.reddit.com/r/MachineLearning/comments/lre3h0/d_how_close_are_operations_research_and_machine/
https://www.splunk.com/en_us/form/5-big-myths-of-ai-and-machine-learning-debunked html?utm_campaign=google_emea_tier2_en_search_generic_observability_it&utm_source=google&utm_medium=cpc&utm_content=5_Big_Myths_AI_ML_EB&utm_term=artificial%20intelligence&device=c&_bt=692788225595&_bm=p&_bn=g&gad_source=1&gbraid=0AAAAAD8kDz2C_jv8Hsj2q5Vd7RZ8kiUnT&gclid=Cj0KCQjwlvW2BhDyARIsADnIe-Irmz7NQaj5JRuBX-jIVgyMomtkBY4NrKC8wQhNSyarQy8mzMHSKY8aAjALE wcB
https://ieor.columbia.edu/operations-research-msor
https://or.ncsu.edu/about-operations-research/what-is-operations-research/

- OR incorporates analytical tools from many different disciplines, so that they can be applied in a rational way to help decision-makers solve problems and control operations in practice
- OR has been taking shape since the industrial revolution, and most notably after WWII

The term Operations Research was first coined in 1940 by McClosky and Trefthen in a small town, Bowdsey, of the United Kingdom, in a military context (the Battle of England).

- A model is a simplified, idealized representation of a real object, a real process, or a real system

- In mathematical models, the building blocks are mathematical structures (equations, inequalities, matrices, functions and operators)
- Building a model:



- The best model of a system strikes a practical compromise between being realistic vs understandable and computationally tractable
- Detail $\neq$ Accuracy (not all details are correct nor necessary)
- A too detailed model brings complexity to analyze it and exploit it
- Models need to be realistic and simple

> *"Everything should be made as simple as possible, but not simpler"*
>
> *Albert Einstein*

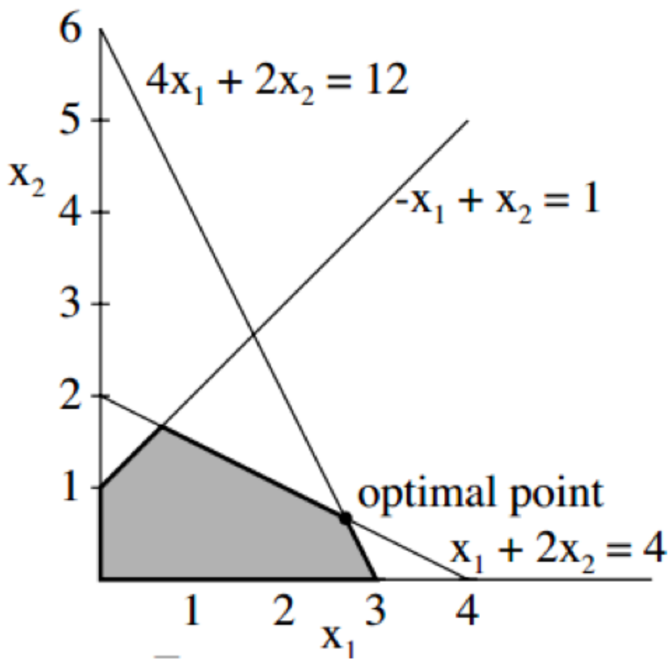
- Does the problem need to be solved?
- What is the *real* problem?
- Would the eventual solution to the problem useful for somebody?
- Would anybody try to implement the solution?
- How much of the analyst's time and cost is worth?
- Are there time and resources available to solve the problem?
- Will the solution create other serious problems for which there is no apparent remedy?

1.2 Problem formulation in Operational Research

Decision variables In what are we going to base our decision?

Objective function How is our function build with respect to the variables?

Constraints What are the limits in our model variables and function?



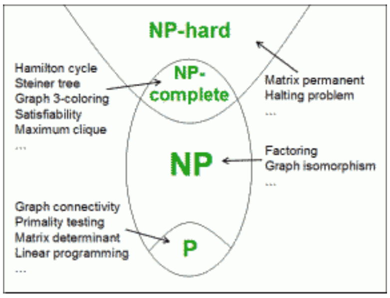
1.3 Algorithms

Sequence of operations that can be carried out in a finite amount of time

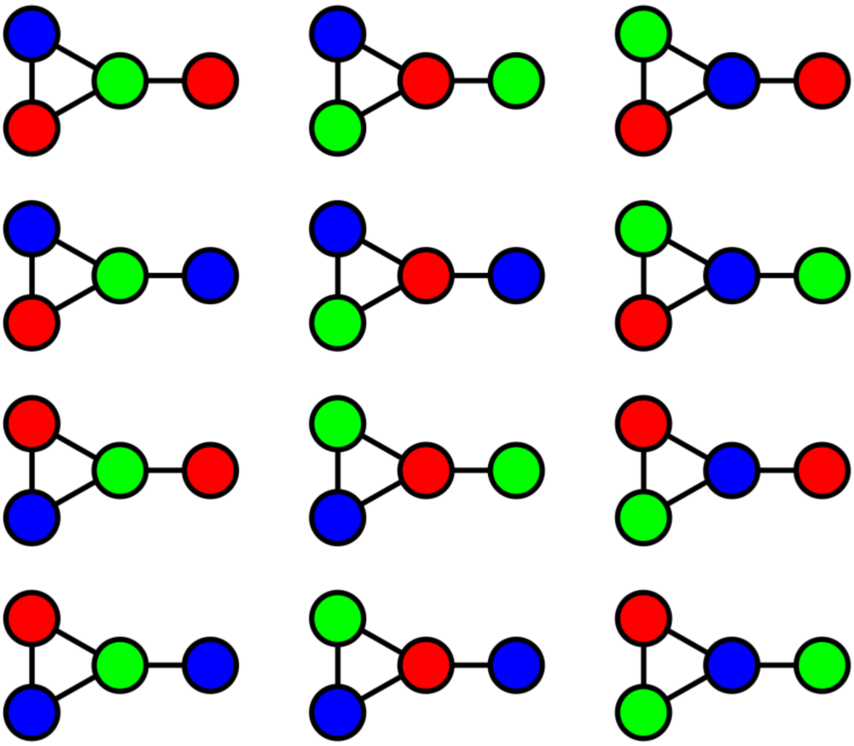
- It can be repeated or called recursively but it will eventually terminate.
- Factors influencing the execution time (unrelated to the algorthm itself):
  - the programming language,
  - the programmer's skills,
  - the hardware being used,
  - the task load on the computer system during execution.
- The performance of an algorithm is typically linked to the size of the problem.

  Class P Problems that can be solved by an algorithm within an amount of computation time proportional to some polynomial function of the problem size.

  Class NP Problems that require computation time proportional to some exponential (or larger) function of the problem size. (The subset sum problem, for example, is $O(2^n \cdot n)$)

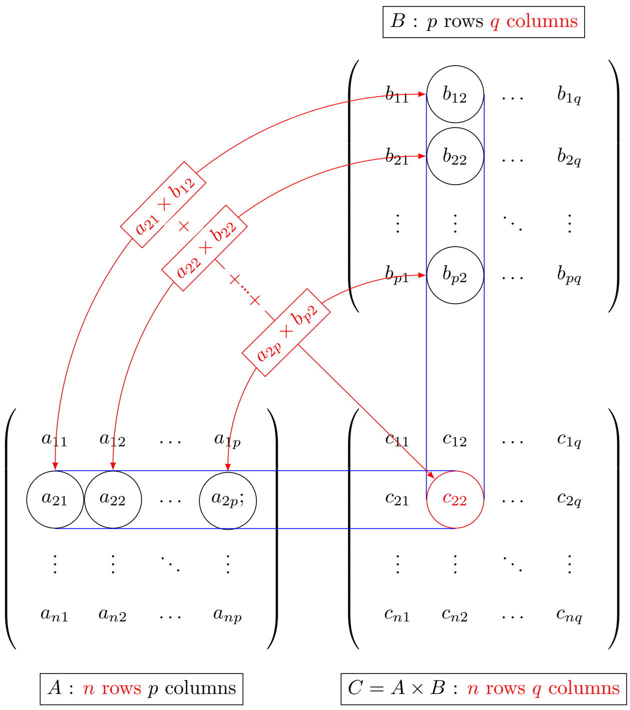


`http://www.solipsys.co.uk/new/PVsNP.html`



- Performance of algorithm independent of software/hardware/developer skill....? # of computational steps.

- Consider worst case scenario: the largest number of steps that may be necessary.



Multiplying two $n \times n$ matrices, in worst case, involves time proportional to $2n^3$.

1.4 Matrix multiplication is $O(n^3)$

```
# this code is contributed by shivanisinghss2110
# https://www.geeksforgeeks.org/strassens-matrix-multiplication/

def multiply(A, B, C):

        for i in range(N):

                for j in range( N):

                        C[i][j] = 0
                        for k in range(N):

                                C[i][j] += A[i][k]*B[k][j]
```

1.5 $O(n)$ notation

- We say $f(n) = O(g(n))$, and we read as "$f(n)$ is big-O of $g(n)$", if there is some C and N such that

$$|f(n)| \leq Cg(n), \ \forall n \geq N.$$

  In the previous example, multiplying two $n \times n$ matrices costs $2n^3 = O(n^3)$ flops[2]. We say the algorithm is of order 3.

- n denotes the problem size and $g(n)$ is some function of problem size.

[2]FLOating Point operationS per second

- $g(n)$ is the algorithm's worst case step count, as a function of n.

- c is the constant of proportionality, and accounbts for extraneous factors affecting execution time (hardware speed, programming style, computer system load during execution)

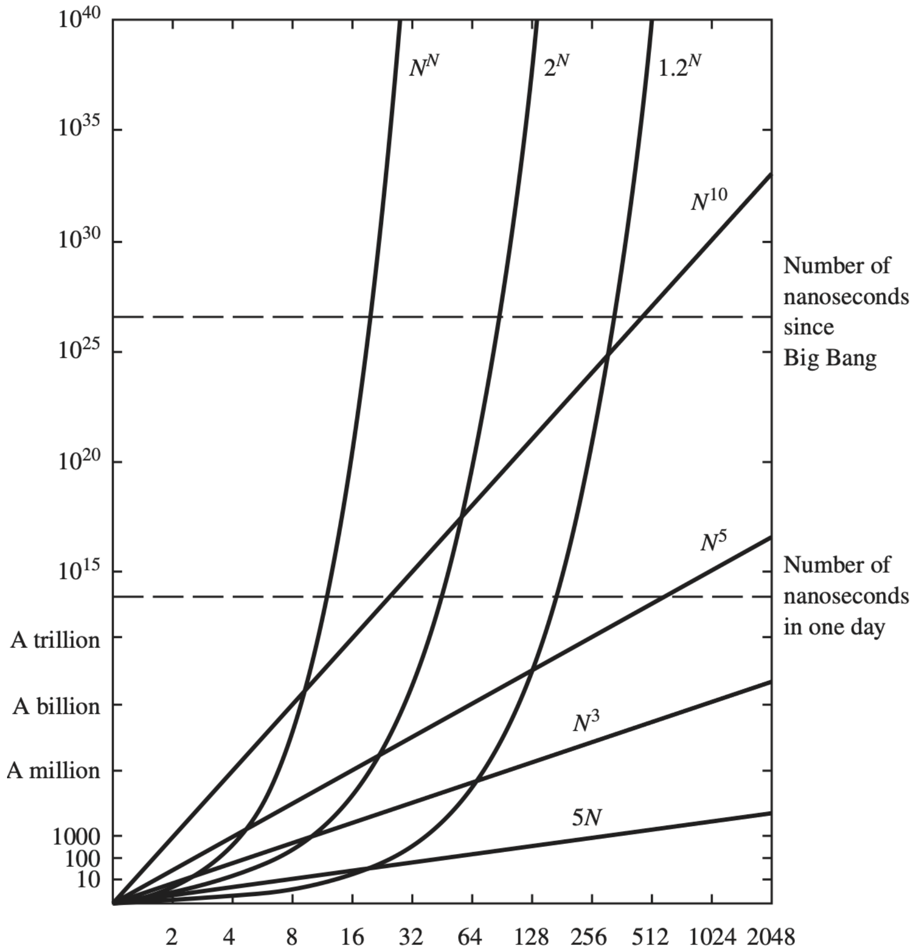
1.6 Different orders, different complexity

	Function \ N	20	60	100	300	1000
Polynomial	$5N$	100	300	500	1500	5000
	$N \times \log_2 N$	86	354	665	2469	9966
	N^2	400	3600	10,000	90,000	1 million (7 digits)
	N^3	8000	216,000	1 million (7 digits)	27 million (8 digits)	1 billion (10 digits)
Exponential	2^N	1,048,576	a 19-digit number	a 31-digit number	a 91-digit number	a 302-digit number
	$N!$	a 19-digit number	an 82-digit number	a 161-digit number	a 623-digit number	unimaginably large
	N^N	a 27-digit number	a 107-digit number	a 201-digit number	a 744-digit number	unimaginably large

Extracted from [1]

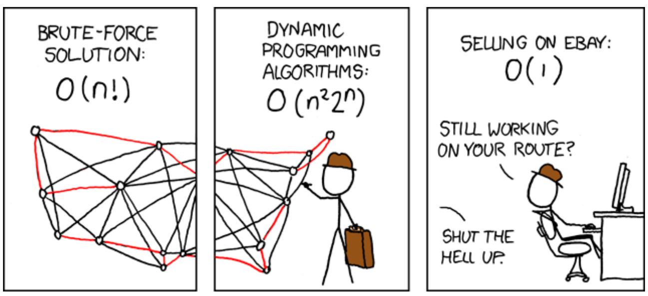
1.7 Growth rate of some functions

For comparison: the number of protons in the known universe has 79 digits; the number of nanoseconds since the Big Bang has 27 digits.[1]



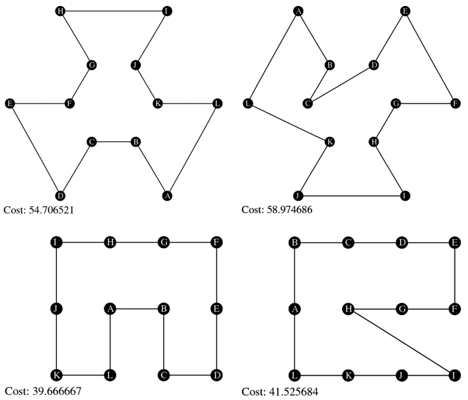
1.8 Search space complexity: the travelling salesman problem (TSP)

- A salesman must visit n cities
- Each city must be visited just once
- Which path should he take to minimize the total distance travelled?



https://xkcd.com/399

1.9 Search space complexity: the travelling salesman problem (TSP)



https://doi.org/10.1073/pnas.0609910104

1.10 TSP practical example

Let us assume there are 4 cities and these are the "distances"[3] between them:

D_{ij}	city A	city B	city C	city D
city A	0	5	10	20
city B	10	0	35	20
city C	20	10	0	5
city D	10	5	10	0

Check code at GitHub

1.11 TSP practical example

- The number of combinations is $(n-1)!$. This is the search space
- Asssuming a computer takes a milisecond to evaluate a solution, How long would it take for the computer to find the optimal solution in our 4 nodes problem? and how long for problems with 10, 20, 50 cities?

1.12 Optimality and Practicality

- We have been trained in mathematics to find exact/perfect solutions. This is not always possible:
  - Models are approximate representations of the real systems
  - Accumulated round-off errors in computers
  - Input data is usually approximated
  - Exponential time algorithms require suboptimal solutions
- "good enough" is not always lowering expectations, as the real world can be extremely complex.

[3]Note that the distance is not necessarily identical in both directions

1.13 Software for Operations Research

Modeling environments Spreadsheets (Excel, Numbers, Google), AMPL, MPL, LINGO, OPL, AIMMS, SAS/OR, OPTMODEL,GAMS, NEOS[4], ...

Solvers Gurobi, Frontline Solver, CPLEX, ...

Software libraries Google OR-Tools, COIN-OR, IMSL, ...

We will be using Google colab through the course when possible.

1.14 Introduction to Operational Research

- Learn about the origins and applications of Operations Research
- Understand system modelling principles
- Understand algorithm efficiency and problem complexity
- Contrast between the optimality and practicality
- Learn about software for operations Research
- Introduction to the Python/Colab environment

References

[1] David Harel and Yishai A. Feldman. *Algorithmics: The Spirit of Computing.* Pearson Education, 2004. Google-Books-ID: txxLovFWkCUC.

[4] check the TSP demo in NEOS/GAMS