AI1: Topic 1 : Week 4

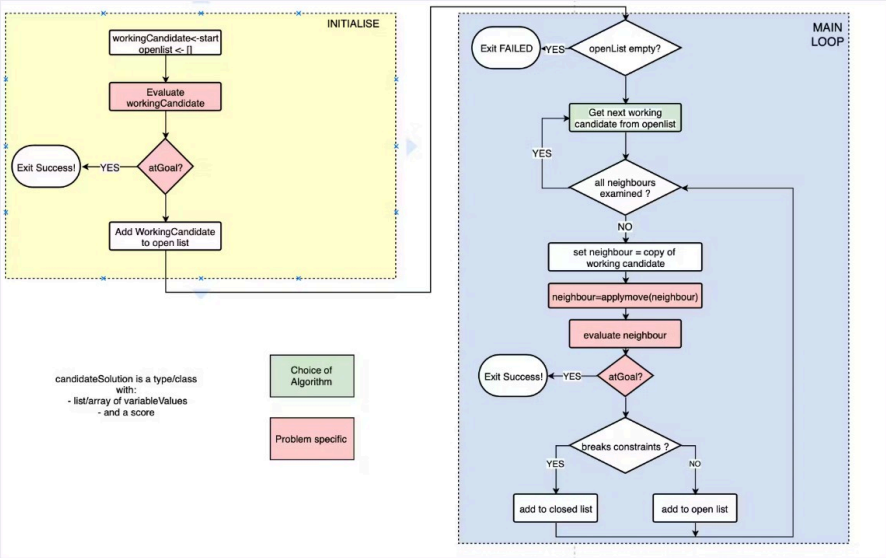
Search in continuous spaces

Attendance Code:

any questions about the module so far?

This week

- Recap questions and discussions
- Search in continuous spaces
- Round up of topic

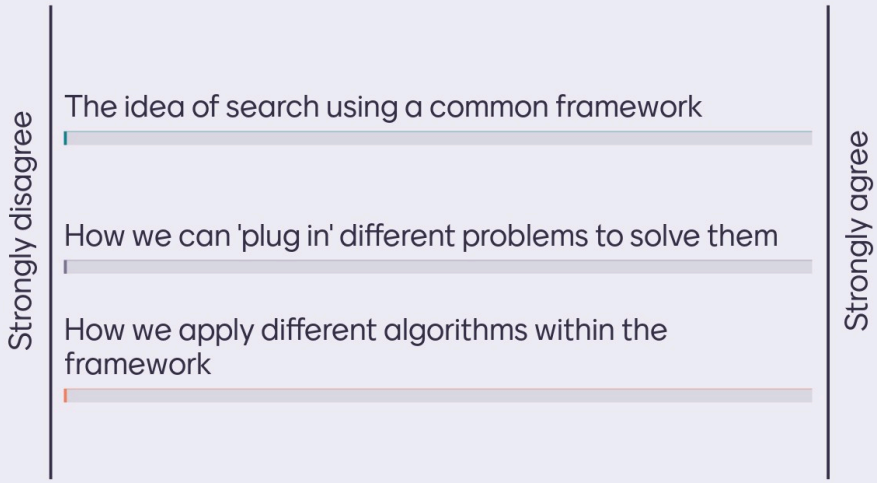


flowchart for common search framework

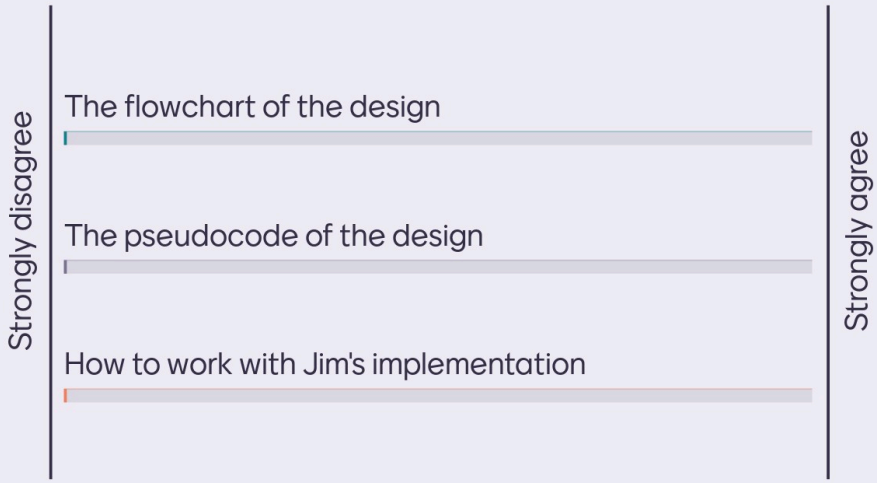
```
INITIALISE
    Set open_list, closed_list ← EmptyList
    working_candidate ← Initialise (CandidateSolution)
    Test ( working_candidate)      Problem-specific code
    AppendToOpenList(working_candidate)

MAIN LOOP
  WHILE IsNotEmpty( open_list) DO
    working_candidate ← SelectAndMoveFromOpenList(algorithm_name)      Algorithm-specific code
    FOR sample in SAMPLE_SIZE DO
      GENERATE
        neighbour ← ApplyMoveOperator (working_candidate)      Representation-specific code
      TEST
        status ← Test ( neighbour)      Problem-specific code
      UPDATE WORKING MEMORY
        IF status IS AtGoal THEN
          Return(SUCCESS)
        ELSE IF status IS BREAKS_CONSTRAINTS THEN
          AppendToClosedList(neighbour)
        ELSE
          AppendToOpenList(neighbour)
    AppendToClosedList(workingCandidate)
```

pseudocode for framework

I understand



I understand



How do we select from the open list?

- Depth-first: Age (youngest first)
- Breadth-first: Age: (oldest first)
- Local Search (hill-climbing): quality first, discard rest
- Best-first: quality first, keep rest
- A*: : cost to reach +quality
- Dijjkstra: Cost to reach

Which of these are *guaranteed* to find the optimal solution?

Hill-Climbers can get stuck in local optima

What do you understand by "optimal"?

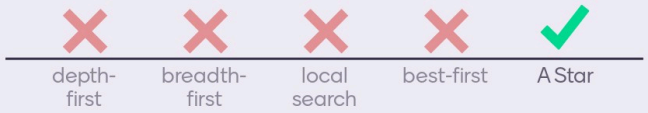
Choosing the right algorithm for a problem

The next few questions describe some scenarios.
In each, select which algorithms you think it *might* be appropriate to apply

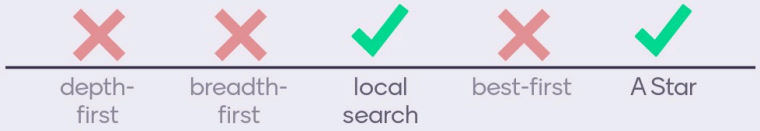
A ‘password-cracking’ problem

Scheduling production in a local cider factory?

A npc planning a path to chase someone in a game?



Organising daily delivery routes ?



Leaderboard

No results yet

Top Quiz participants will be displayed here once there are results!

UWE Bristol | University of the West of England

I'd expect you to be able to ...

- recognise what algorithm is being applied if I showed you code
- tell me what line defined the algorithm
- characterise algorithms in terms of efficiency, completeness, optimality
- characterise the **context** of a problem in terms of the trade-offs between completeness/optimality/efficiency
- select the most suitable **algorithm(s)** for a simple problem-solving scenario
- select the most suitable **representation** for a simple problem,

Searching in different types of spaces

- A candidate solution encodes values for a set of decisions that define a solution
- examples so far: decisions take **categorical** values
- so at every stage we have a fixed set of neighbours
- **What happens if the decisions take real/continuous values?**

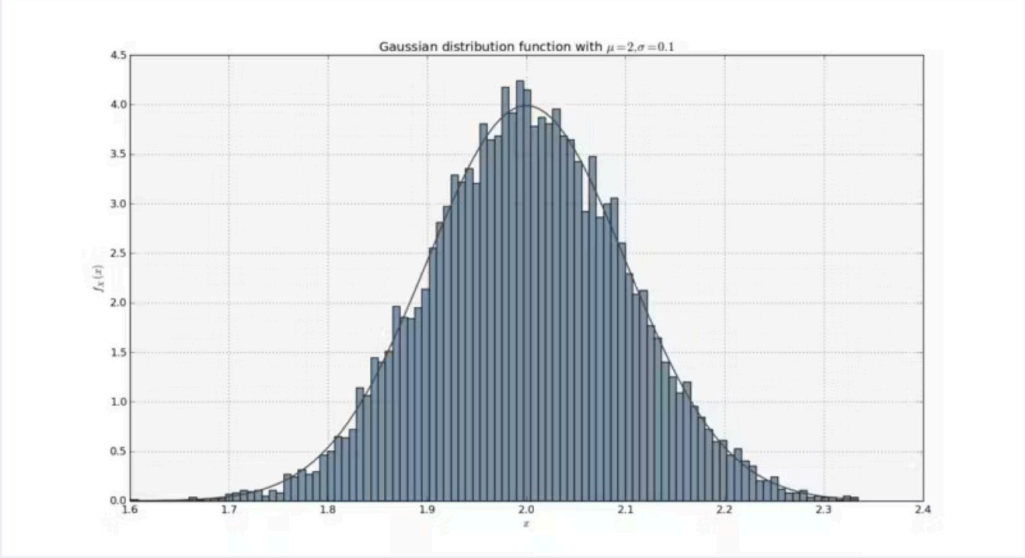
Continuous variables have infinite numbers of neighbours!

only limited by precision of floating points
so we can't examine all the neighbours.

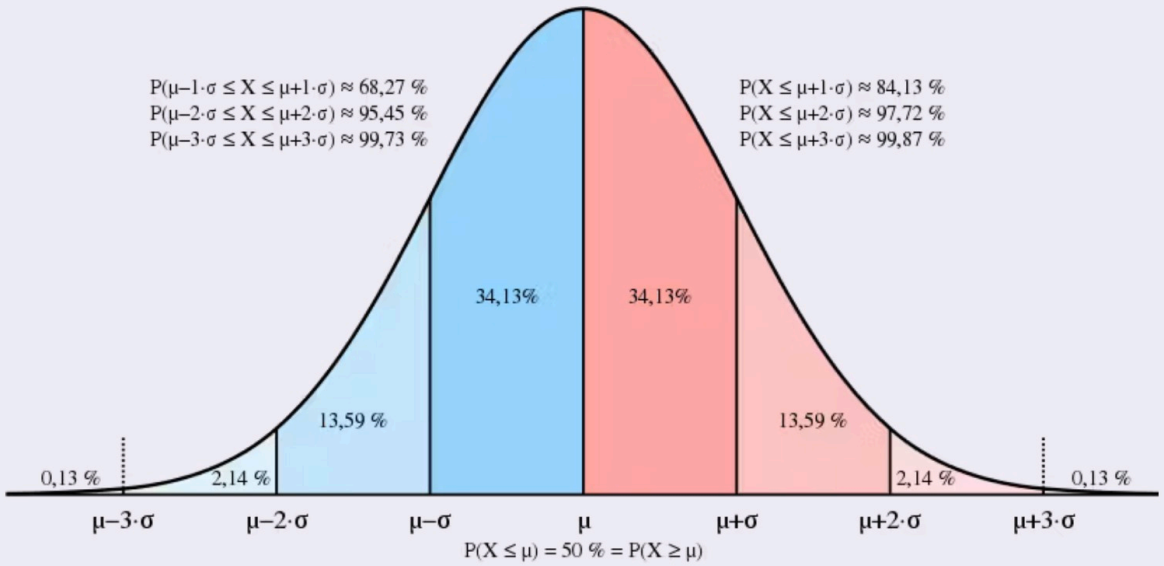
Option1: make neighbours by adding noise and
just sample a few

Option 2: put. the effort into use some cunning
maths, and just generate one neighbour

Can any one describe what a Gaussian/Normal distribution of numbers is



normal distribution from samples Mean is called mu (greek letter like a u) and standard deviation is called sigma (like an o with a tail)



Probability of generating points from a random distribution

UWE Bristol | University of the West of England

Option 1: Adding noise and sampling

- **Zero-mean** : change up as likely as change down
- **Standard Deviation** (sigma) choose to suit scale of problem
- For each variable:
- * generate a random number from a N(0,1) distribution
- * scale to problem (multiply by sigma)
- * add the random numbers to the variable
- Might need some trial and error to decide how many samples to use

option 2: where possible

- → Apply some maths to estimate local slope
- → do this while calculating quality
- → move operator = 1 step in that direction
- → repeat

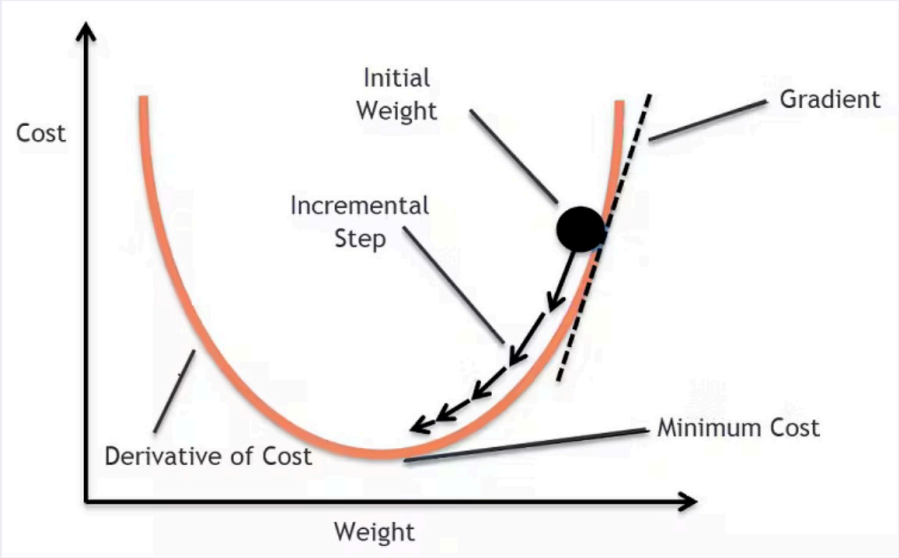


Image from https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/

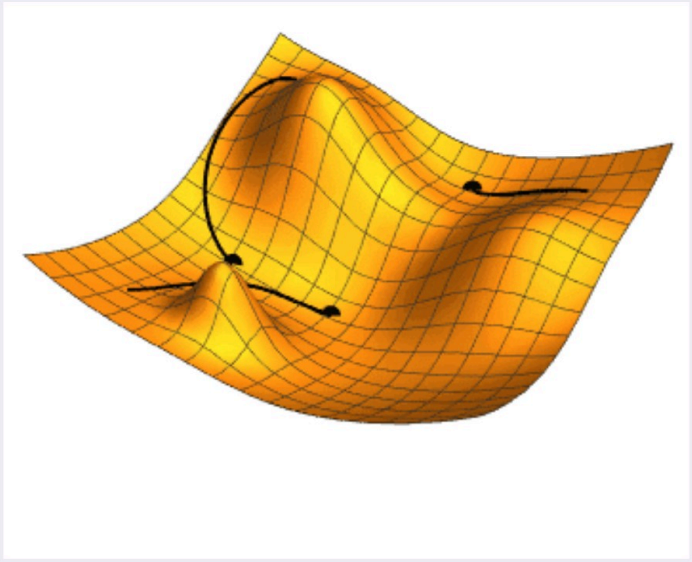


image from Wikimedia

Any questions?