Complex Planning and Optimization Problems

Part 2: Experiment, Experiment, Experiment!

1

Repetition

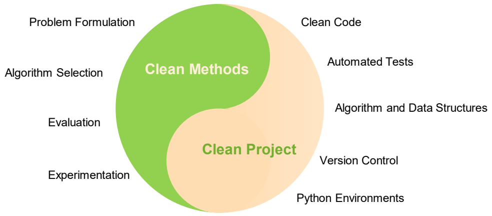
Why are Planning and Optimization similar?

What activities does a Planning/Optimization project usually involve?

Why is it important to challenge your assumptions?

2

Goal of this learning unit: Best Practices


Problem Formulation

Algorithm Selection

Evaluation

Experimentation

Clean Methods

Clean Project

Clean Code

Automated Tests

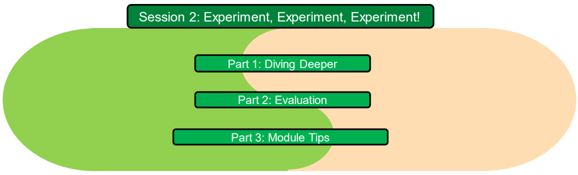
Algorithm and Data Structures

Version Control

Python Environments

3

Sessions

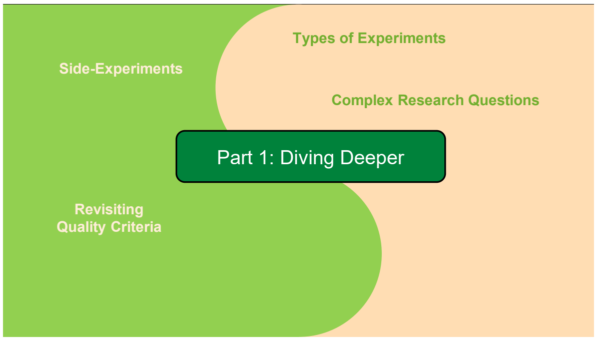

Session 2: Experiment, Experiment, Experiment!

Part 1: Diving Deeper

Part 2: Evaluation

Part 3: Module Tips

4


Types of Experiments

Side-Experiments

Complex Research Questions

Part 1: Diving Deeper

Revisiting Quality Criteria

5

Reminder: We are Detectives

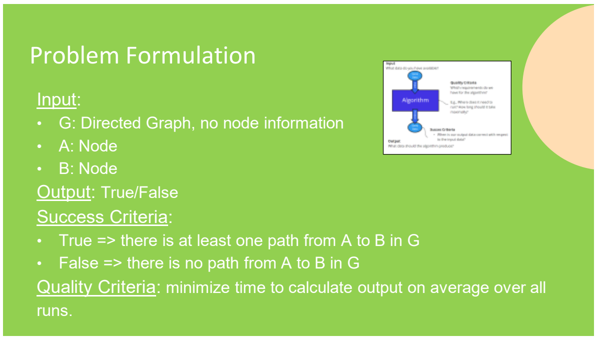
The goal: Try to find out if Person A has been in communication with Person B

The Complication: They may have used a third person to obfuscate their tracks.

The Evidence: We traced emails and know who has been writing to whom.

Side-Goal: Time is of the essence!

6



Problem Formulation

Input
What data do you have available?
Algorithm
Quality Criteria
What requirements do we have for the algorithm?
E.g., Where does it need to run? How long should it take maximally?
Success Criteria
When is our output data correct with respect to the input data?
Output
What data should the algorithm produce?

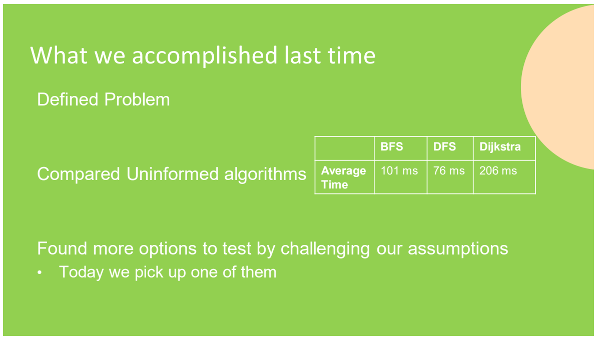
Input:
- G: Directed Graph, no node information
- A: Node
- B: Node

Output: True/False

Success Criteria:
- True => there is at least one path from A to B in G
- False => there is no path from A to B in G

Quality Criteria: minimize time to calculate output on average over all runs.

7



What we accomplished last time

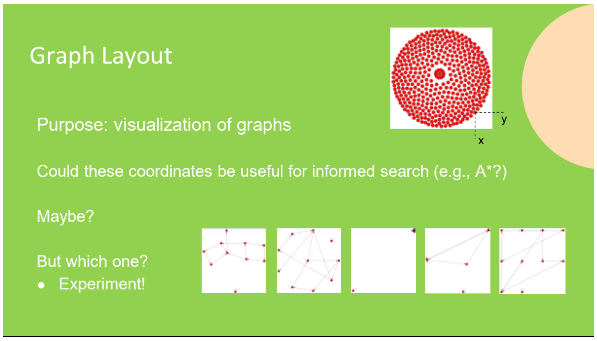
Defined Problem

Compared Uninformed algorithms

	BFS	DFS	Dijkstra
Average Time	101 ms	76 ms	206 ms

Found more options to test by challenging our assumptions
- Today we pick up one of them

8



Graph Layout

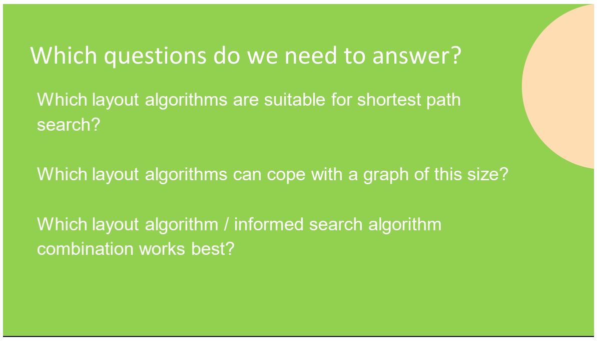
Purpose: visualization of graphs

Could these coordinates be useful for informed search (e.g., A*?)

Maybe?

But which one?
- Experiment!

9



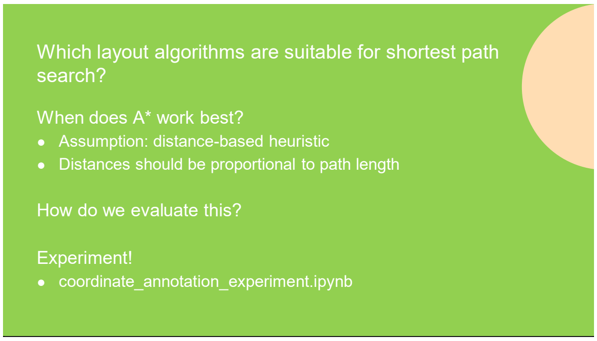
Which questions do we need to answer?

Which layout algorithms are suitable for shortest path search?

Which layout algorithms can cope with a graph of this size?

Which layout algorithm / informed search algorithm combination works best?

10



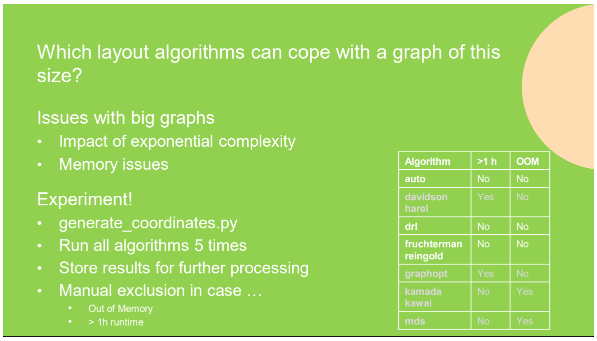
Which layout algorithms are suitable for shortest path search?

When does A* work best?
- Assumption: distance-based heuristic
- Distances should be proportional to path length

How do we evaluate this?

Experiment!
- coordinate_annotation_experiment.ipynb

11



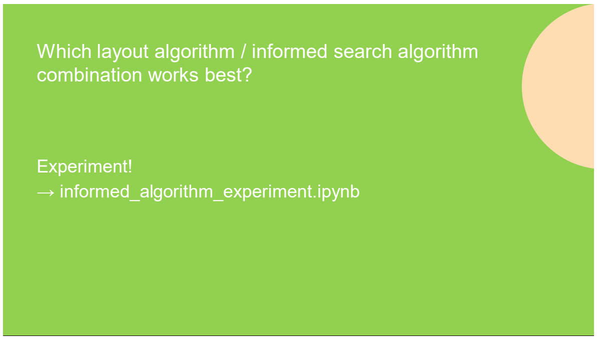
Which layout algorithms can cope with a graph of this size?

Issues with big graphs
- Impact of exponential complexity
- Memory issues

Experiment!
- generate_coordinates.py
- Run all algorithms 5 times
- Store results for further processing
- Manual exclusion in case …
  - Out of Memory
  - > 1h runtime

Algorithm	>1 h	OOM
auto	No	No
davidson harel	Yes	No
drl	No	No
fruchterman reingold	No	No
graphopt	Yes	No
kamada kawai	No	Yes
mds	No	Yes

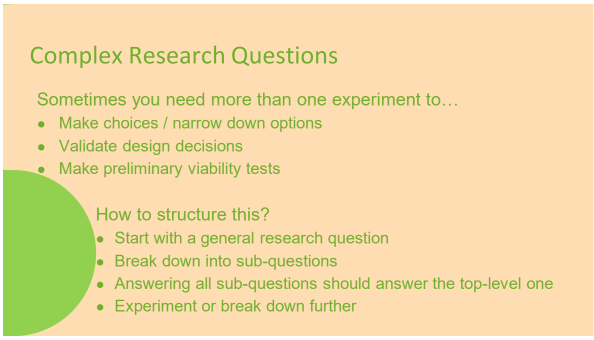
12



Which layout algorithm / informed search algorithm combination works best?

Experiment!
→ informed_algorithm_experiment.ipynb

13



Complex Research Questions

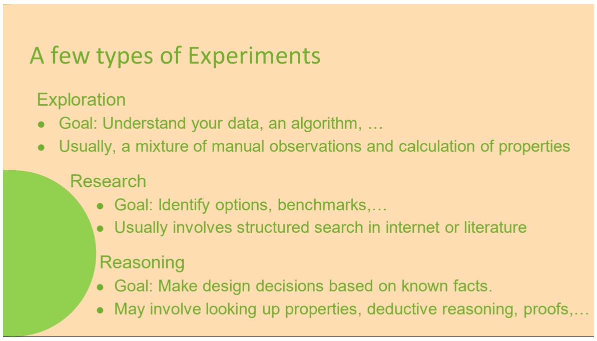
Sometimes you need more than one experiment to…

- Make choices / narrow down options
- Validate design decisions
- Make preliminary viability tests

How to structure this?

- Start with a general research question
- Break down into sub-questions
- Answering all sub-questions should answer the top-level one
- Experiment or break down further

14



A few types of Experiments

Exploration

- Goal: Understand your data, an algorithm, …
- Usually, a mixture of manual observations and calculation of properties

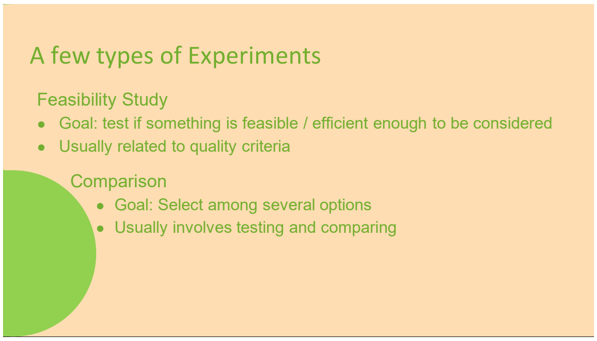
Research

- Goal: Identify options, benchmarks,…
- Usually involves structured search in internet or literature

Reasoning

- Goal: Make design decisions based on known facts.
- May involve looking up properties, deductive reasoning, proofs,…

15



A few types of Experiments

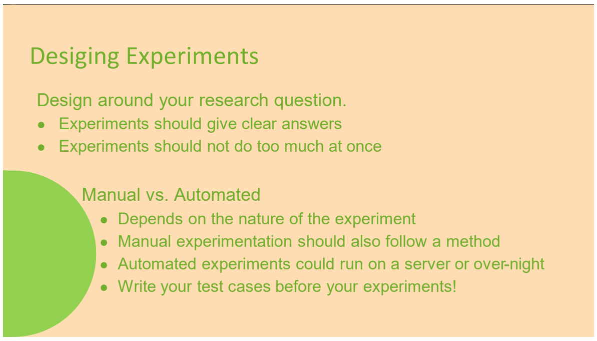
Feasibility Study

- Goal: test if something is feasible / efficient enough to be considered
- Usually related to quality criteria

Comparison

- Goal: Select among several options
- Usually involves testing and comparing

16



Desiging Experiments

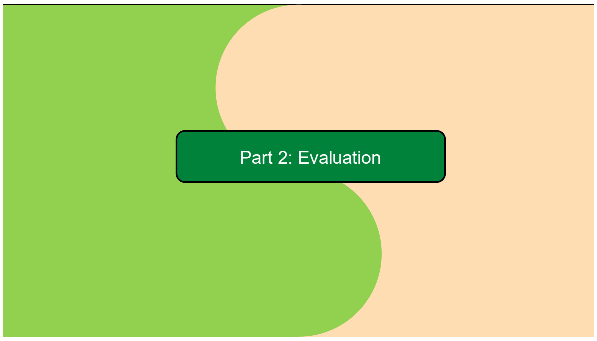
Design around your research question.

- Experiments should give clear answers
- Experiments should not do too much at once

Manual vs. Automated

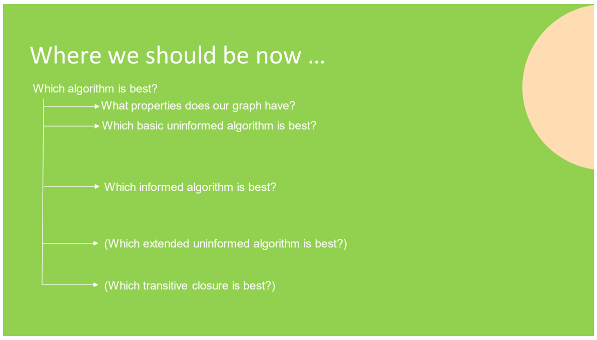
- Depends on the nature of the experiment
- Manual experimentation should also follow a method
- Automated experiments could run on a server or over-night
- Write your test cases before your experiments!

17



Part 2: Evaluation

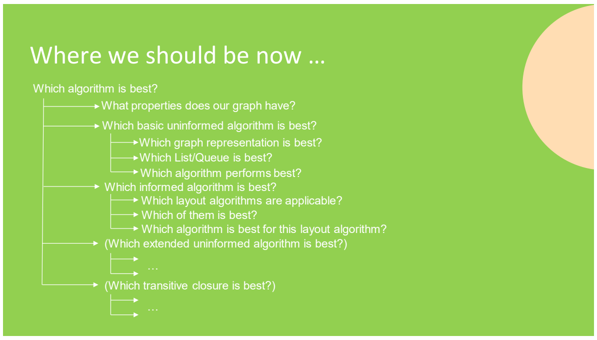
18



Where we should be now …

Which algorithm is best?
- What properties does our graph have?
- Which basic uninformed algorithm is best?
- Which informed algorithm is best?
- (Which extended uninformed algorithm is best?)
- (Which transitive closure is best?)

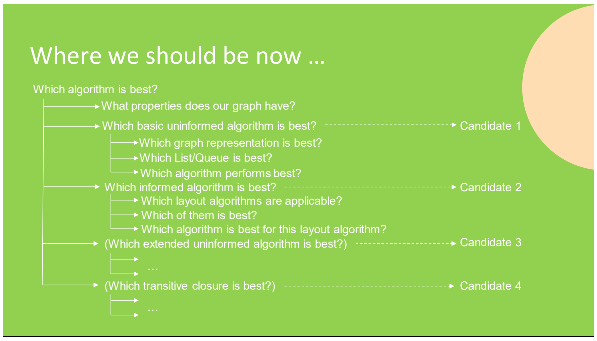
19



Where we should be now …

Which algorithm is best?
- What properties does our graph have?
- Which basic uninformed algorithm is best?
  - Which graph representation is best?
  - Which List/Queue is best?
  - Which algorithm performs best?
- Which informed algorithm is best?
  - Which layout algorithms are applicable?
  - Which of them is best?
  - Which algorithm is best for this layout algorithm?
- (Which extended uninformed algorithm is best?)
  - …
- (Which transitive closure is best?)
  - …

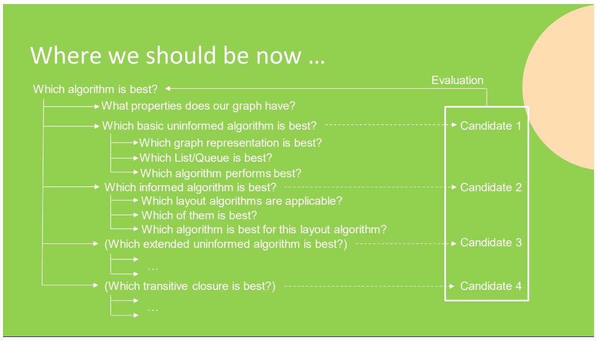
20



Where we should be now …

Which algorithm is best?
- What properties does our graph have?
- Which basic uninformed algorithm is best? → Candidate 1
  - Which graph representation is best?
  - Which List/Queue is best?
  - Which algorithm performs best?
- Which informed algorithm is best? → Candidate 2
  - Which layout algorithms are applicable?
  - Which of them is best?
  - Which algorithm is best for this layout algorithm?
- (Which extended uninformed algorithm is best?) → Candidate 3
  - …
- (Which transitive closure is best?) → Candidate 4
  - …

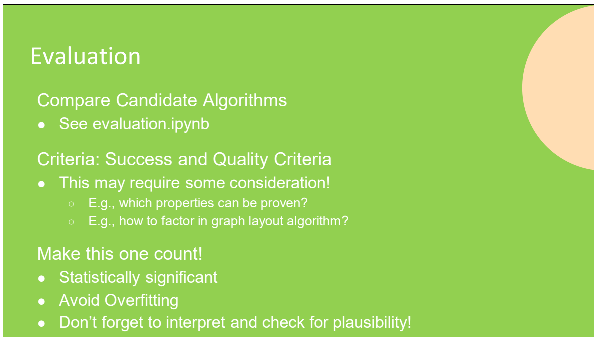
21



Where we should be now …

Which algorithm is best? ← Evaluation
- What properties does our graph have?
- Which basic uninformed algorithm is best? → Candidate 1
  - Which graph representation is best?
  - Which List/Queue is best?
  - Which algorithm performs best?
- Which informed algorithm is best? → Candidate 2
  - Which layout algorithms are applicable?
  - Which of them is best?
  - Which algorithm is best for this layout algorithm?
- (Which extended uninformed algorithm is best?) → Candidate 3
  - …
- (Which transitive closure is best?) → Candidate 4
  - …

22



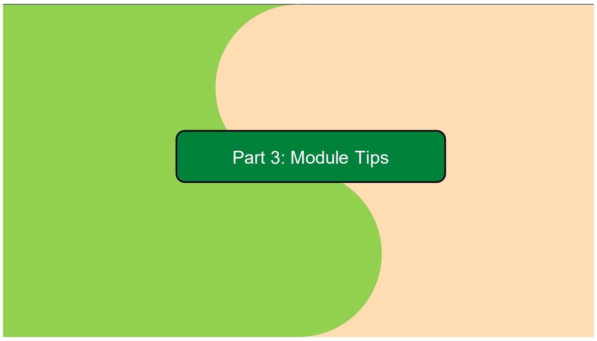
Evaluation

Compare Candidate Algorithms
- See evaluation.ipynb

Criteria: Success and Quality Criteria
- This may require some consideration!
  - E.g., which properties can be proven?
  - E.g., how to factor in graph layout algorithm?

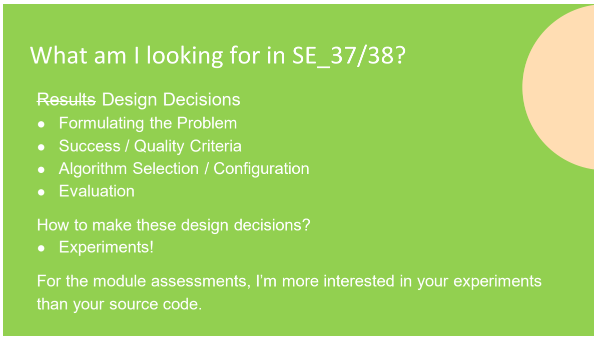
Make this one count!
- Statistically significant
- Avoid Overfitting
- Don't forget to interpret and check for plausibility!

23



Part 3: Module Tips

24



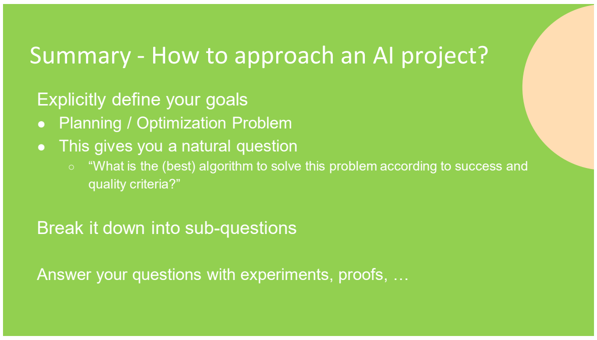
What am I looking for in SE_37/38?

~~Results~~ Design Decisions
- Formulating the Problem
- Success / Quality Criteria
- Algorithm Selection / Configuration
- Evaluation

How to make these design decisions?
- Experiments!

For the module assessments, I'm more interested in your experiments than your source code.

25



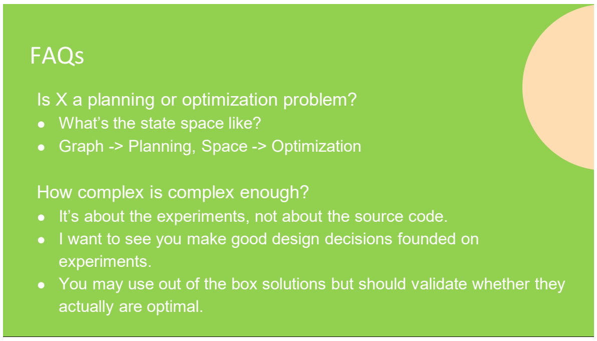
Summary - How to approach an AI project?

Explicitly define your goals
- Planning / Optimization Problem
- This gives you a natural question
  - "What is the (best) algorithm to solve this problem according to success and quality criteria?"

Break it down into sub-questions

Answer your questions with experiments, proofs, …

26

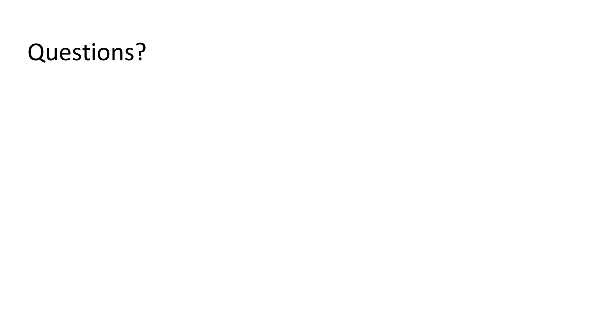


FAQs

Is X a planning or optimization problem?
- What's the state space like?
- Graph -> Planning, Space -> Optimization

How complex is complex enough?
- It's about the experiments, not about the source code.
- I want to see you make good design decisions founded on experiments.
- You may use out of the box solutions but should validate whether they actually are optimal.

27



Questions?

28