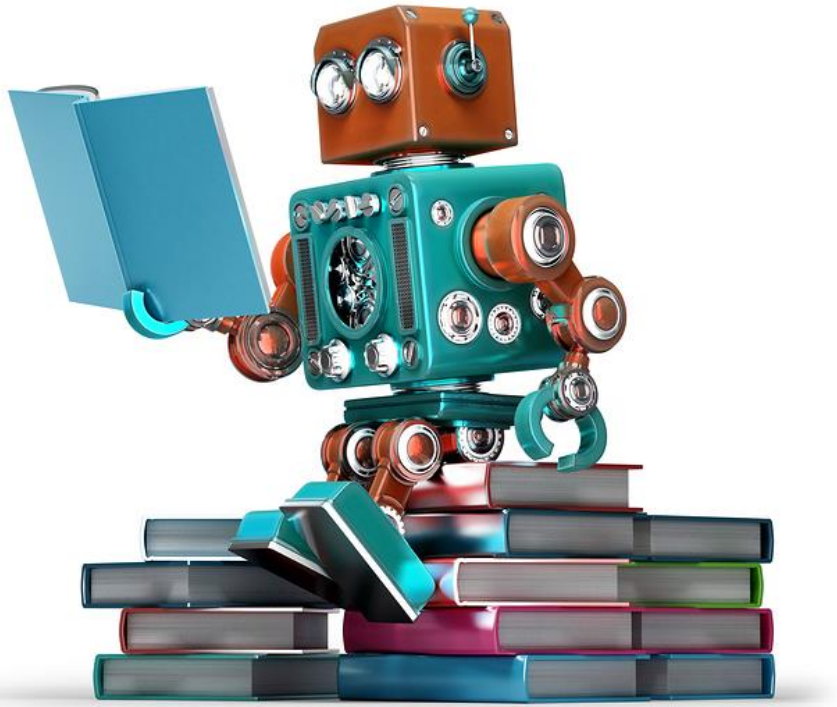
REASONING SYSTEMS
DAY 1



https://robohub.org/wp-content/uploads/2016/11/bigstock-Retro-Robot-Reading-A-Book-Is-110707406.jpg

DAY 1 AGENDA

1.1 Reasoning Systems Overview

1.2 Uninformed Search Techniques

1.3 Search Representation Workshop

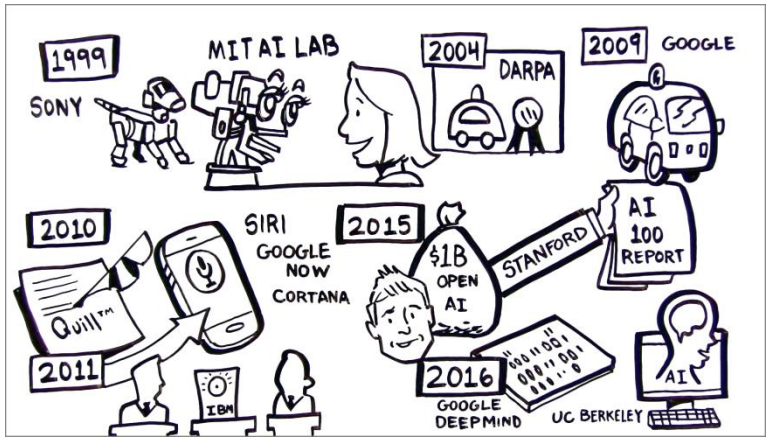
DAY 1 TIMETABLE

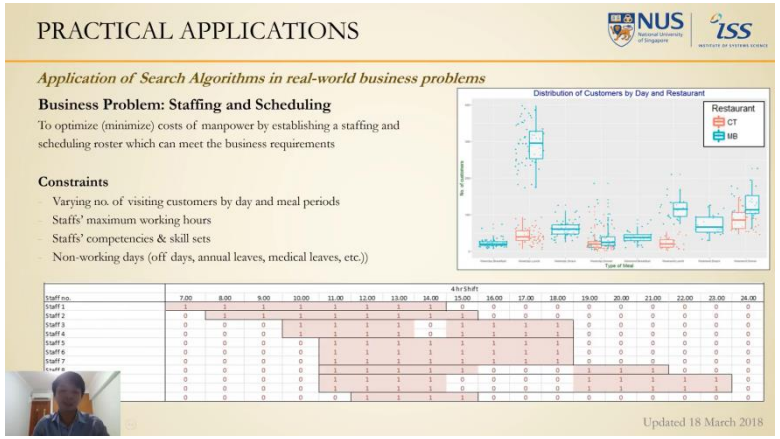
No	Time	Topic	By Whom	Where
1	9 am	Welcome and Introduction	GU Zhan (Sam)	Class
2	9.30 am	1.1 Reasoning Systems Overview	GU Zhan (Sam)	Class
3	10.10 am	Morning Break		
4	10.30 am	1.2 Uninformed Search Techniques	GU Zhan (Sam)	Class
5	12.10 pm	Lunch Break		
6	1.30 pm	1.3 Search Representation Workshop 1	All	Class
7	3.10 pm	Afternoon Break		
8	3.30 pm	1.3 Search Representation Workshop 2	All	Class
9	4.50 pm	Summary and Review	All	Class
10	5 pm	End		

1.1
REASONING SYSTEMS OVERVIEW

1.1 REASONING SYSTEMS OVERVIEW

- **AI is a grand reasoning system.**





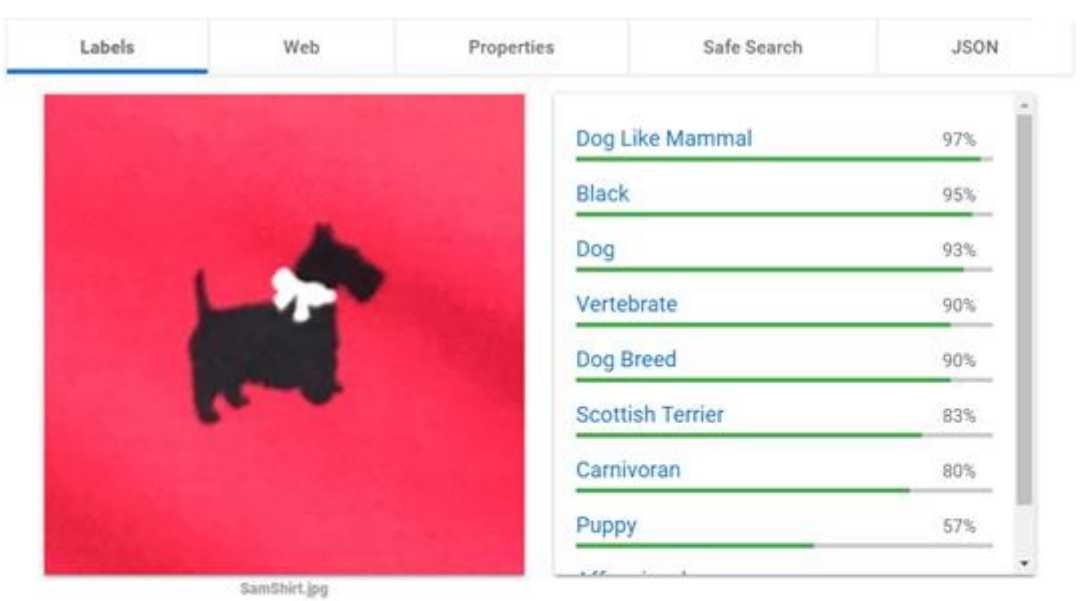
1.1 REASONING SYSTEMS OVERVIEW

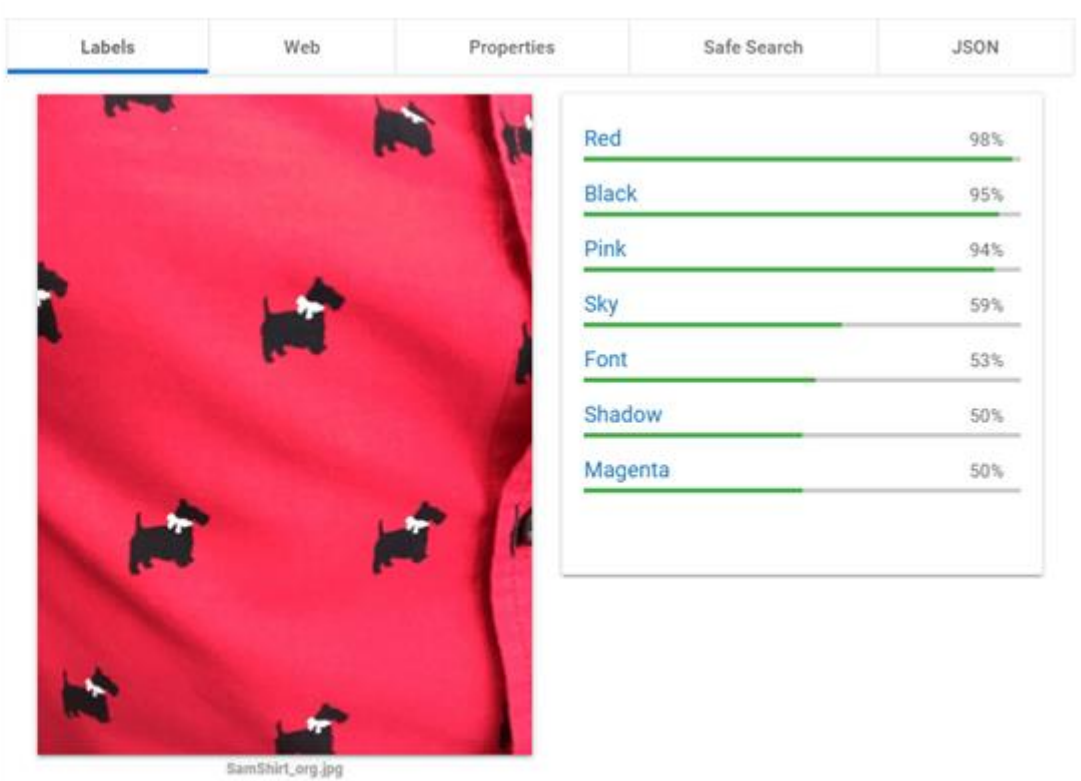
- **Question Answering System: IBM Watson**

IBM Watson

1.1 REASONING SYSTEMS OVERVIEW

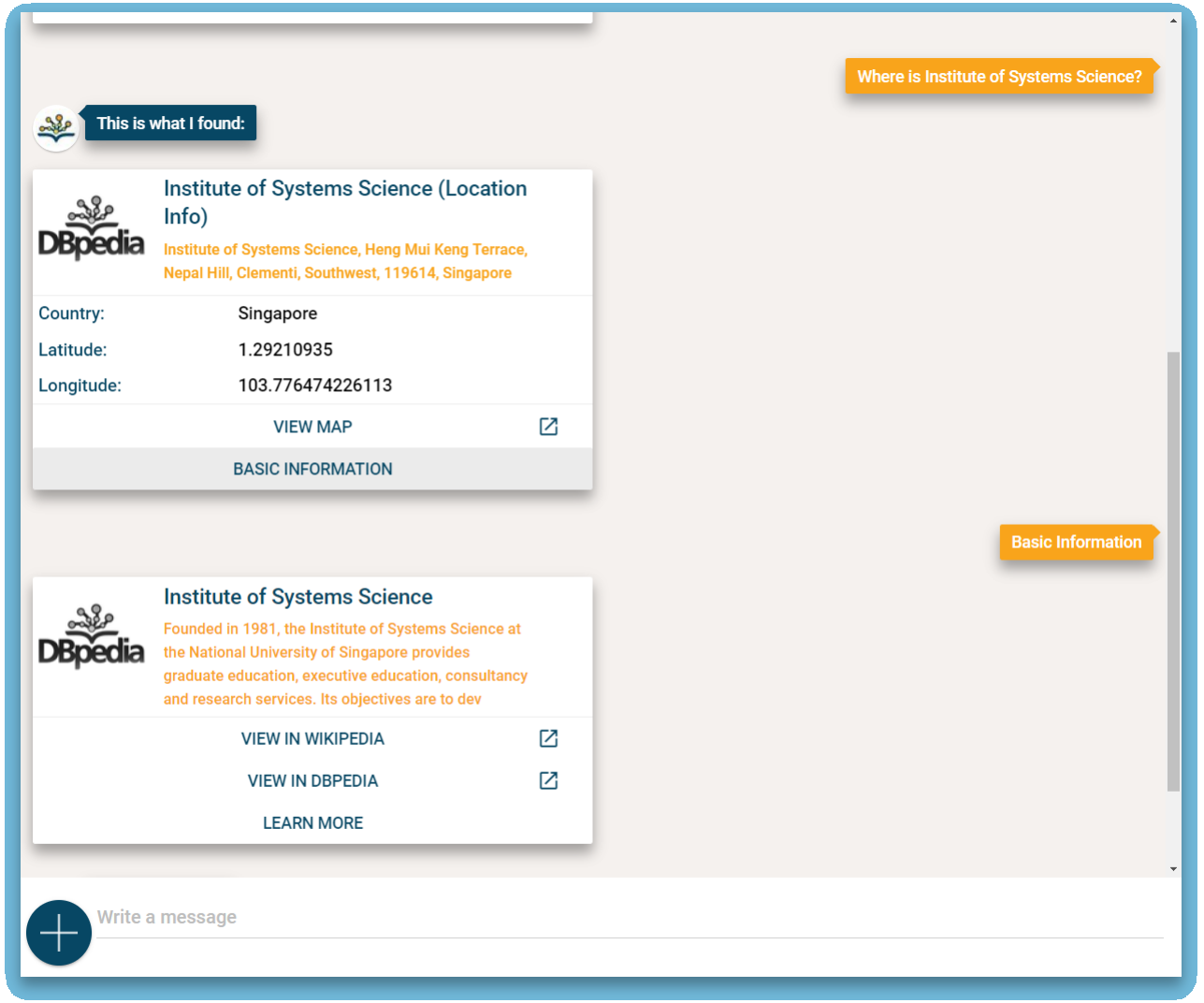
- Image Object Recognition: Google Vision





1.1 REASONING SYSTEMS OVERVIEW

- **Chat-Bot: DBpedia**

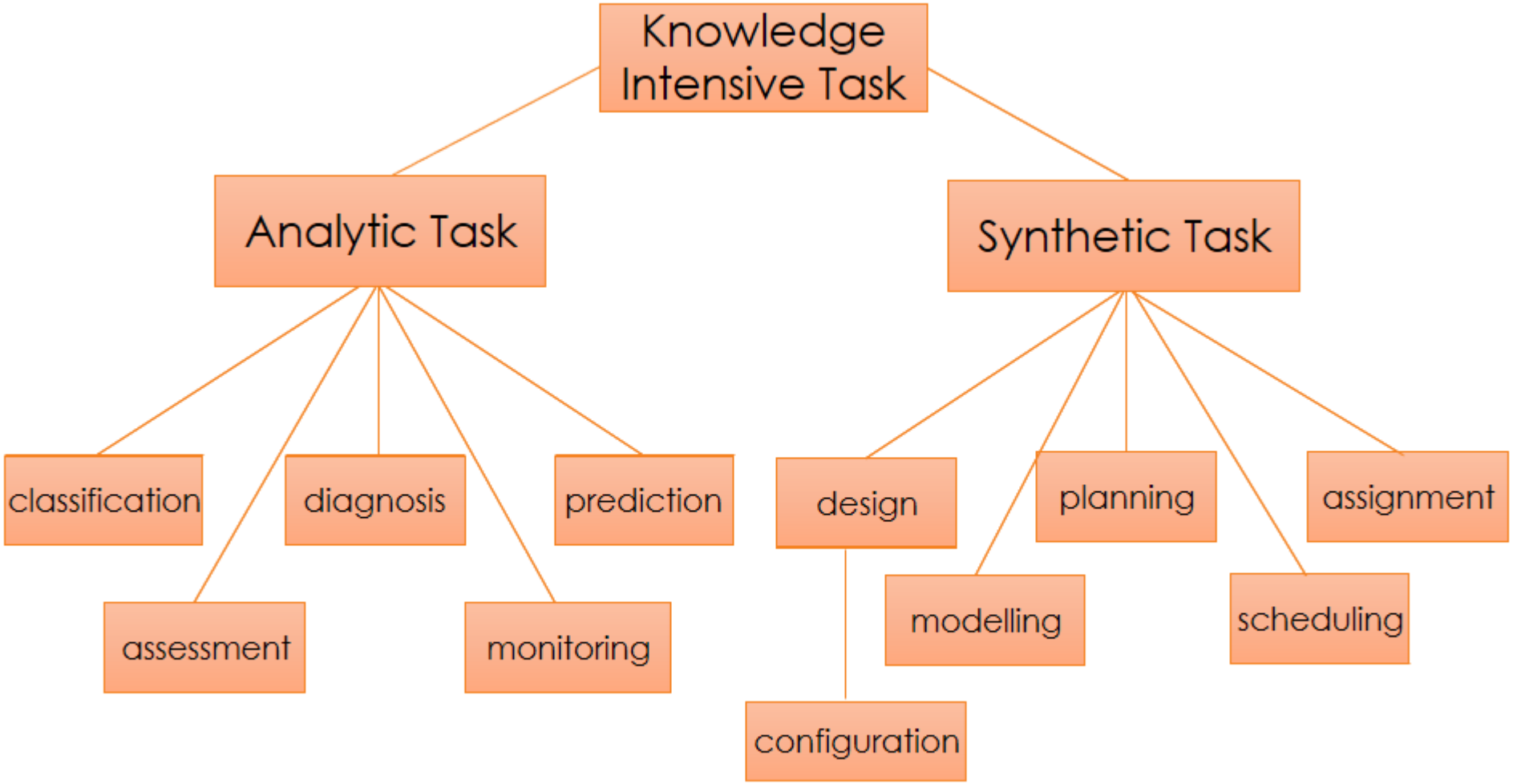


1.1 REASONING SYSTEMS OVERVIEW

- **Reasoning AI** — **IRS → Machine Reasoning**
  Deductive & Abductive reasoning & decision making
- **Thinking AI** — **ISA = Intelligent Software Agents**
  Make predictions; Take strategic actions
- **Learning AI** — **PRS = Pattern Recognition Systems**
  Inductive reasoning; Statistical machine learning
- **Cognitive AI** — **IRS → Cognitive Systems**
  Learn and reason with purpose; Interact with humans
- **Problem Solving AI** — **IRS → Reasoning Systems**
  Plan; Optimize; Search for solution

1.1 REASONING SYSTEMS OVERVIEW

Problem Solving Task Hierarchy



1.1 REASONING SYSTEMS OVERVIEW

Problem Solving Task Types

- **Analytic Tasks**
  - System/Solution to be analysed pre-exists, but usually not completely "known".
  - Input: some data to trigger the system (e.g. patient symptoms)
  - Output: some characterization or behaviours about the system (e.g. cause of illness)
- **Synthetic Tasks**
  - System/Solution does not yet exist.
  - Input: requirements about system to be constructed
  - Output: constructed system description

1.1 REASONING SYSTEMS OVERVIEW

Problem Solving of Analytic Tasks

- **Analytic Tasks**

  Identification, Classification, Prediction, Clustering/Grouping, …

- **Techniques (Machine Reasoning)**

  Heuristic Business Rules

  Decision Trees

  Case Based Reasoning

  Fuzzy Logic

  Rule Induction

  Machine Learning

  …

1.1 REASONING SYSTEMS OVERVIEW

Problem Solving of Synthetic Tasks

- **Synthetic Tasks**

  Planning, Scheduling, Optimisation, Design, …

- **Techniques (Reasoning Systems)**

  Uninformed (brute force / blind) Search

  Informed (heuristic) Search

  Simulations

  Genetic Algorithms

  Reinforcement Learning

  Data Mining

  …

1.2
UNINFORMED SEARCH TECHNIQUES

1.2 UNINFORMED SEARCH TECHNIQUES

- Solving Problem by Search
- Search Tree Representation
- Depth First Search (DFS)
- Breadth First Search (BFS)

https://modernmarketingtoday.com/wp-content/uploads/2013/02/search-marketing.jpg

1.2 UNINFORMED SEARCH TECHNIQUES

Solving Problem by Search

- Synthesis of a new valid solution is performed by searching through the **(search/solution) space**, which contains all possible solutions
- Each possible solution is evaluated to see whether it is **valid** and/or the **optimum** (best solution found by now), e.g. a valid employee schedule, a valid vehicle delivery route, an optimal (shortest) vehicle delivery route,
- Validity of solution involves satisfaction of a set of **constraints** on the solution variables
- Optimality is measured by a **user-defined function** which measures the "goodness" of the solution, e.g. the shorter delivery route the better.

1.2 UNINFORMED SEARCH TECHNIQUES

Solving Problem by Search

(1) Create a pool of solution candidates (search space)

(2) Pick up one candidate solution from pool

(3) Check whether this candidate is valid (constraints satisfied?)

- (3)=True If valid, continue
- (3)=False If not valid, go to (2)

(4) Check whether this candidate is the best till now (optimal solution?)

- (4)=True If best, save this solution as the best then continue
- (4)=False If not best, discard this solution then continue

(5) Go to (2). Repeat the cycle until a stopping criteria is met.

1.2 UNINFORMED SEARCH TECHNIQUES

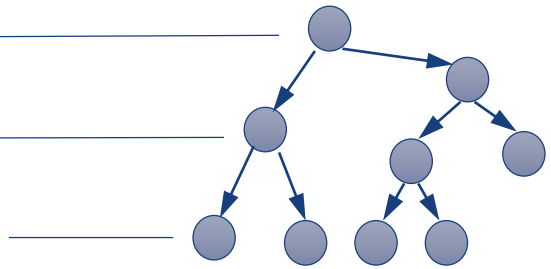
Search Tree Representation

- **Search is illustrated using a search space with a particular restricted structure**
- **Solutions (search space) can be represented as a Tree**
  - Nodes in tree represent

    an initial state

    an intermediate state

    a final state (feasible solution, or failure)



  - Connection between nodes represents a search step

1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

- Always prefers to search deeper in the search tree rather than wider.

1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

Algorithm Pseudo Code

(1) Set **N** to be a list of initial nodes

(2) If **N** is empty, then exit and signal failure

(3) Set **n** to be the first node in **N**, and remove **n** from **N**

(4) Check **n**:

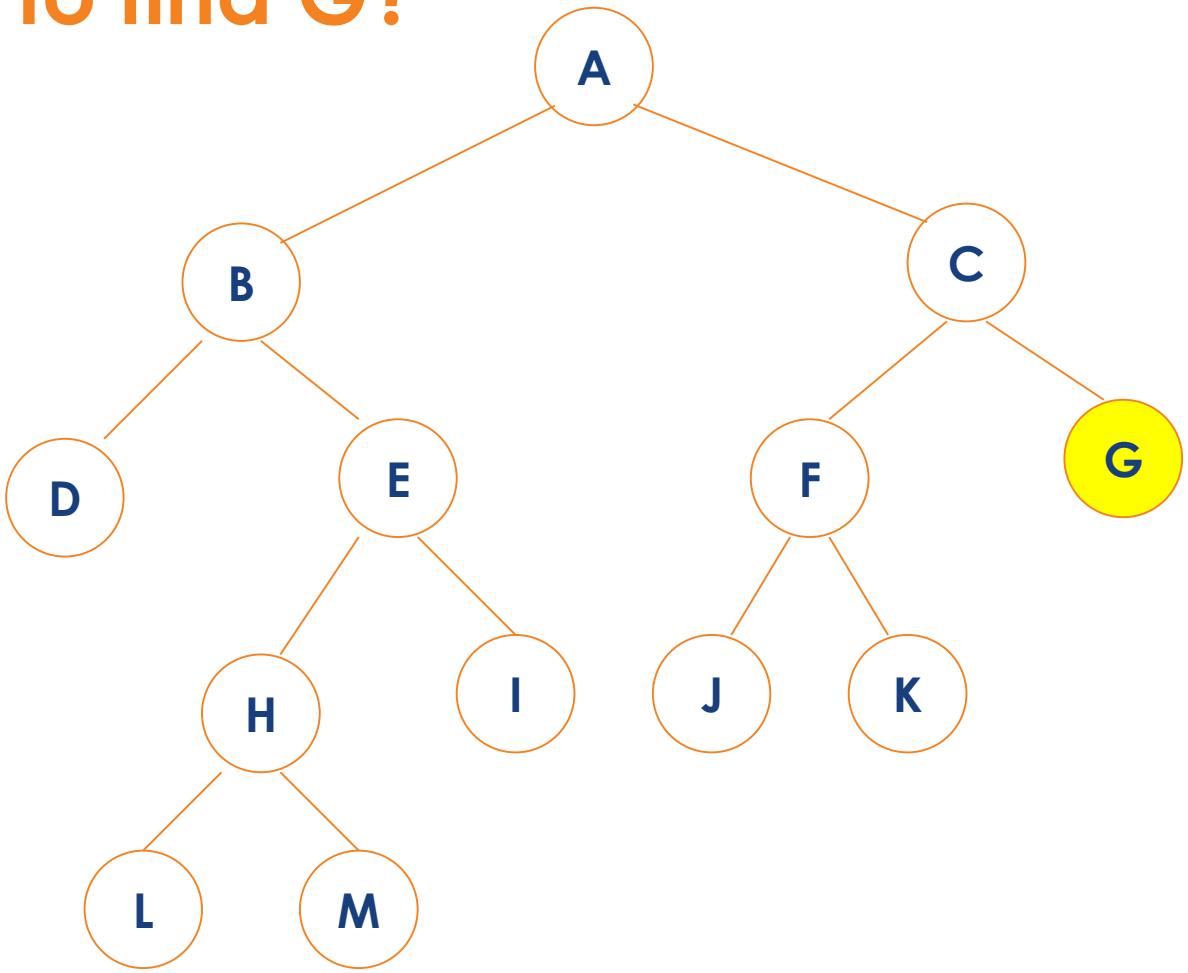
(4.1) If **n** is a goal node, then exit and signal success

(4.2) Otherwise, add the children of **n** to the **front** of **N** then go to step **(2)**

1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

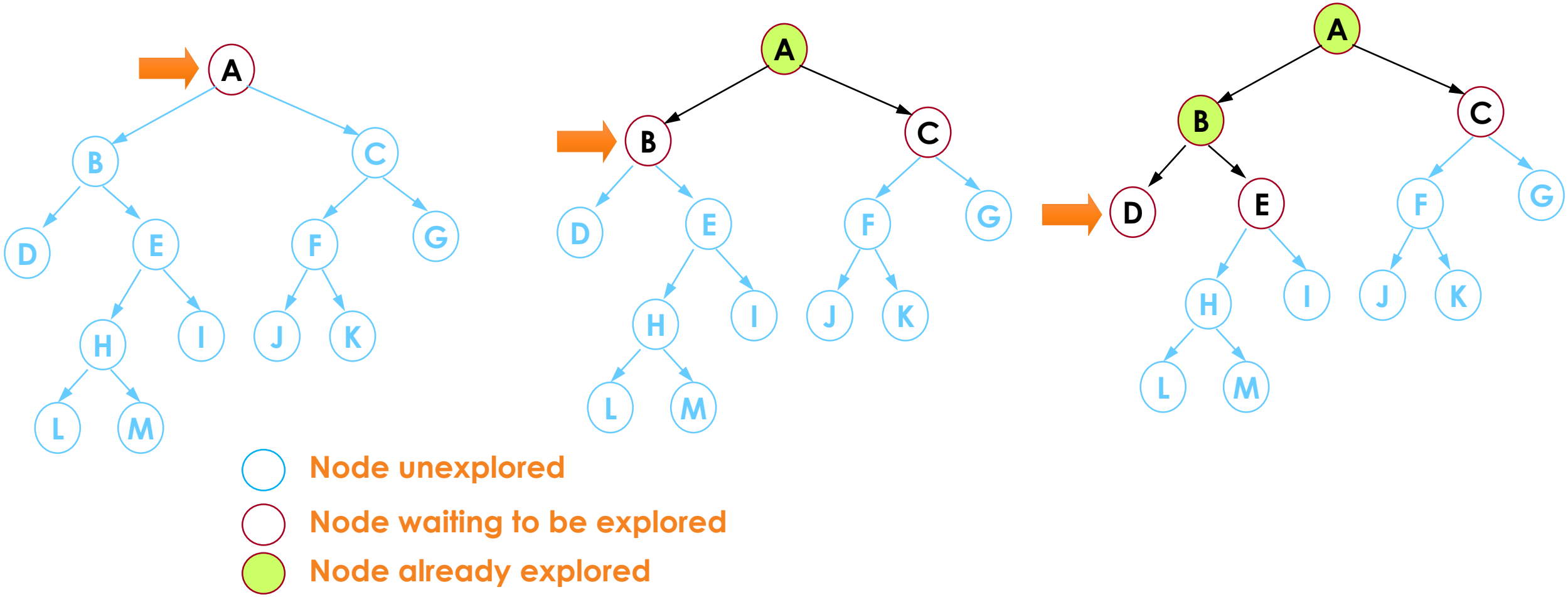
- Visit order to find G?



1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

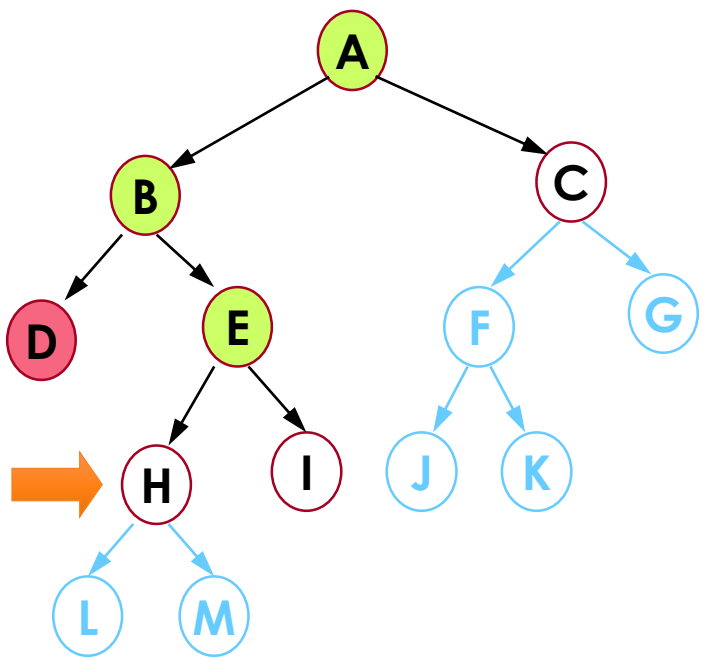
- Keep track of nodes

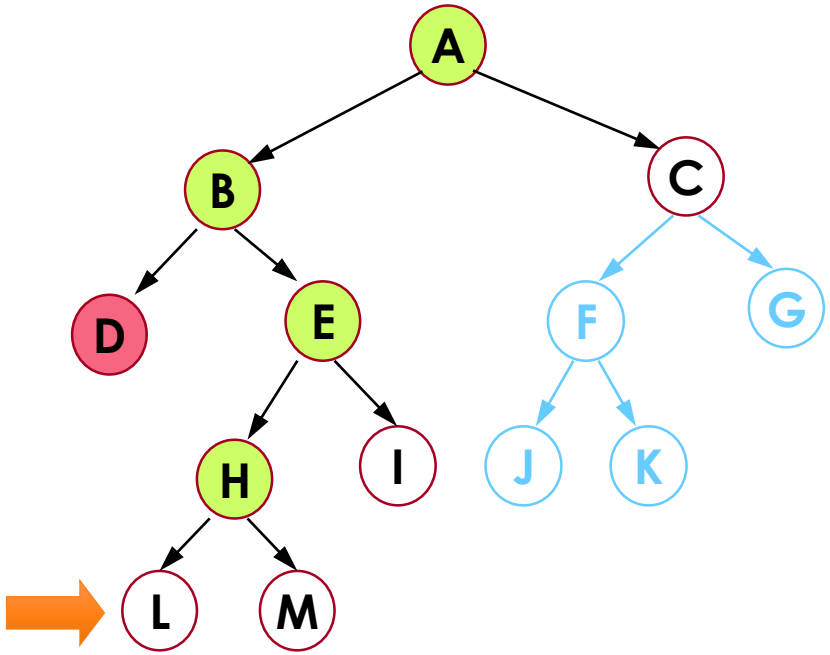


1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

- Keep track of nodes



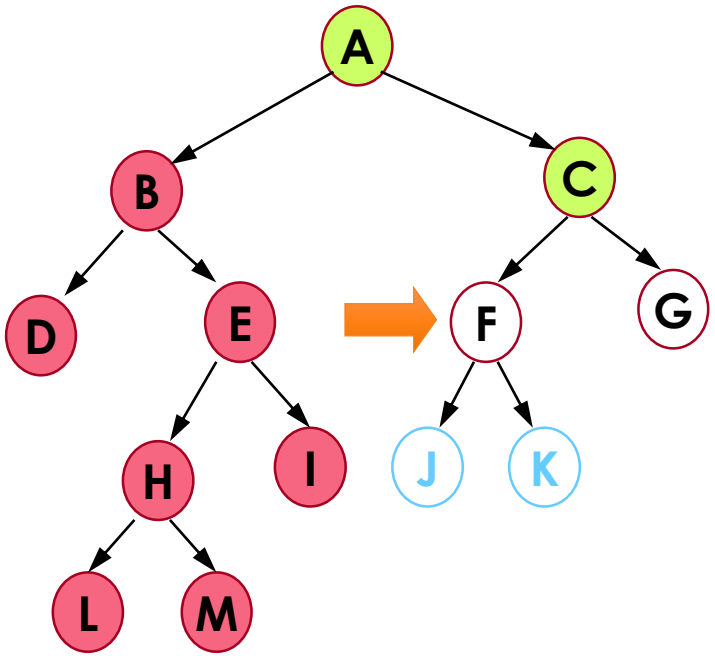


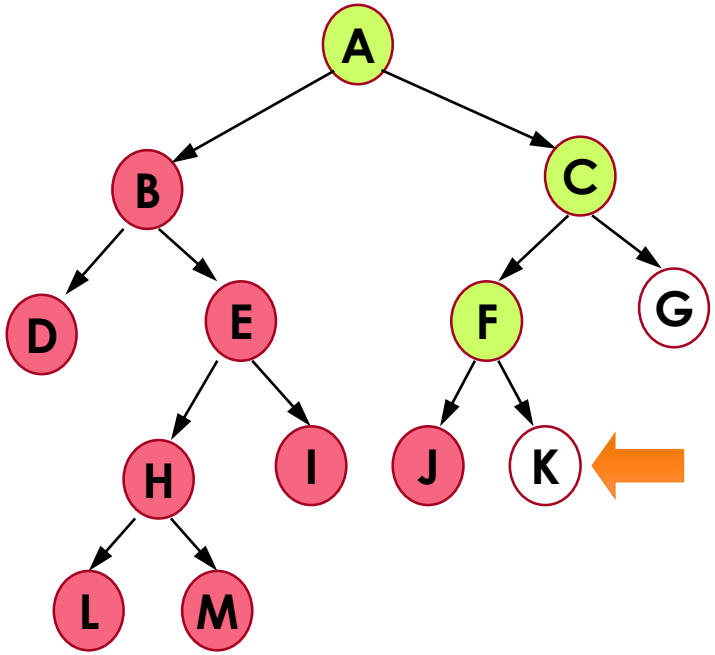
Explored non-solution node can be removed

1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

- Keep track of nodes



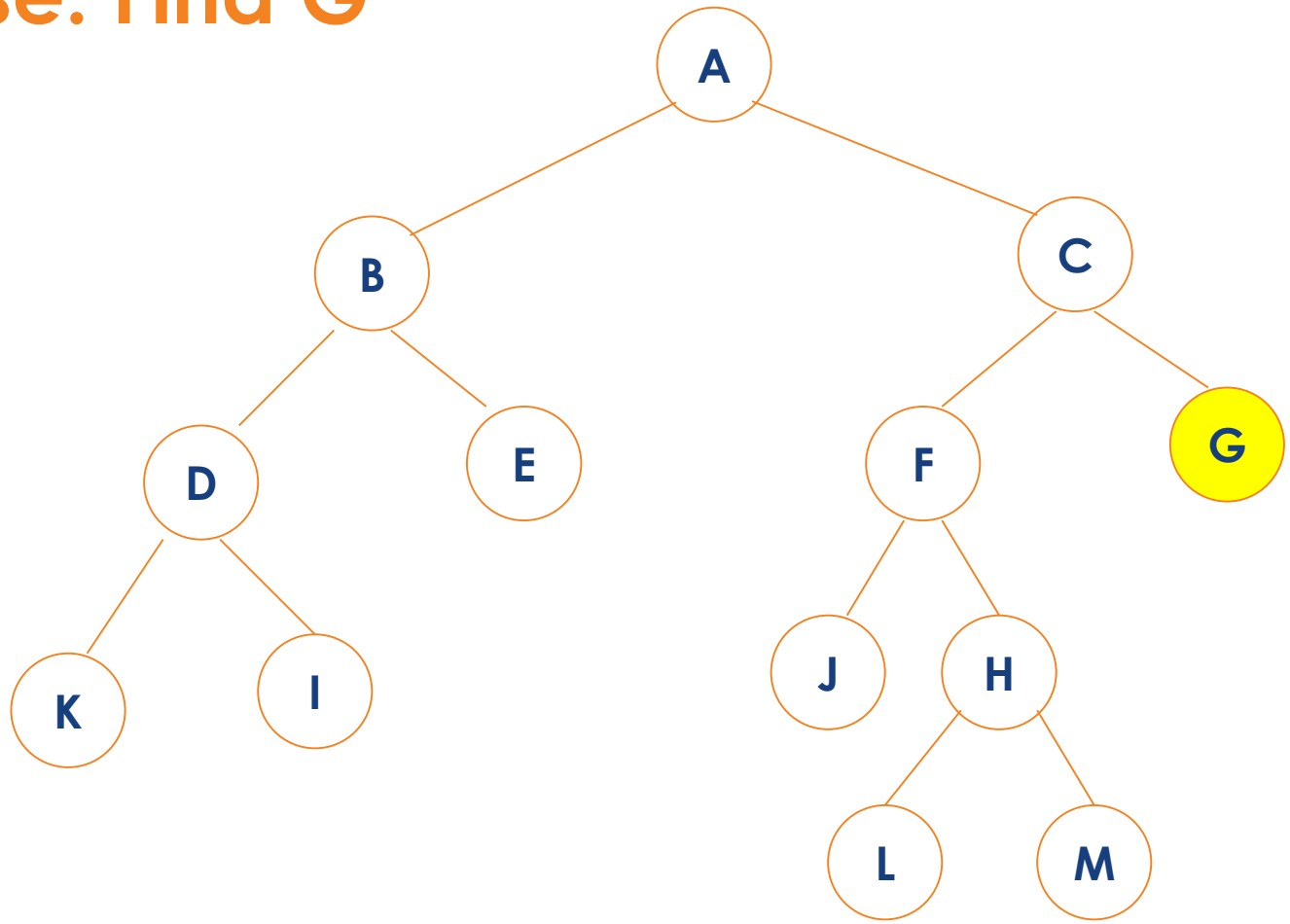


Explored non-solution node/branch can be removed

1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

- **Exercise: Find G**

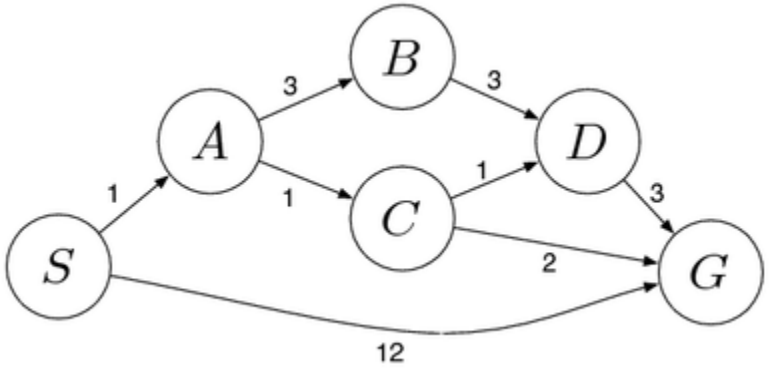


1.2 UNINFORMED SEARCH TECHNIQUES

Depth First Search (DFS)

DFS Drawbacks

- Can make a wrong choice and get stuck going down a very long (or even infinite: loops in graph) path when a different choice would lead to a solution near the root of the search tree.



Source https://algorithmicthoughts.wordpress.com/2012/12/15/artificial-intelligence-uniform-cost-searchucs/

1.2 UNINFORMED SEARCH TECHNIQUES

Breadth First Search (BFS)

- Explores all the nodes at a given depth before processing deeper in the search tree.

1.2 UNINFORMED SEARCH TECHNIQUES

Breadth First Search (BFS)

Algorithm Pseudo Code

(1) Set **N** to be a list of initial nodes

(2) If **N** is empty, then exit and signal failure

(3) Set **n** to be the first node in **N**, and remove **n** from **N**

(4) Check **n**:

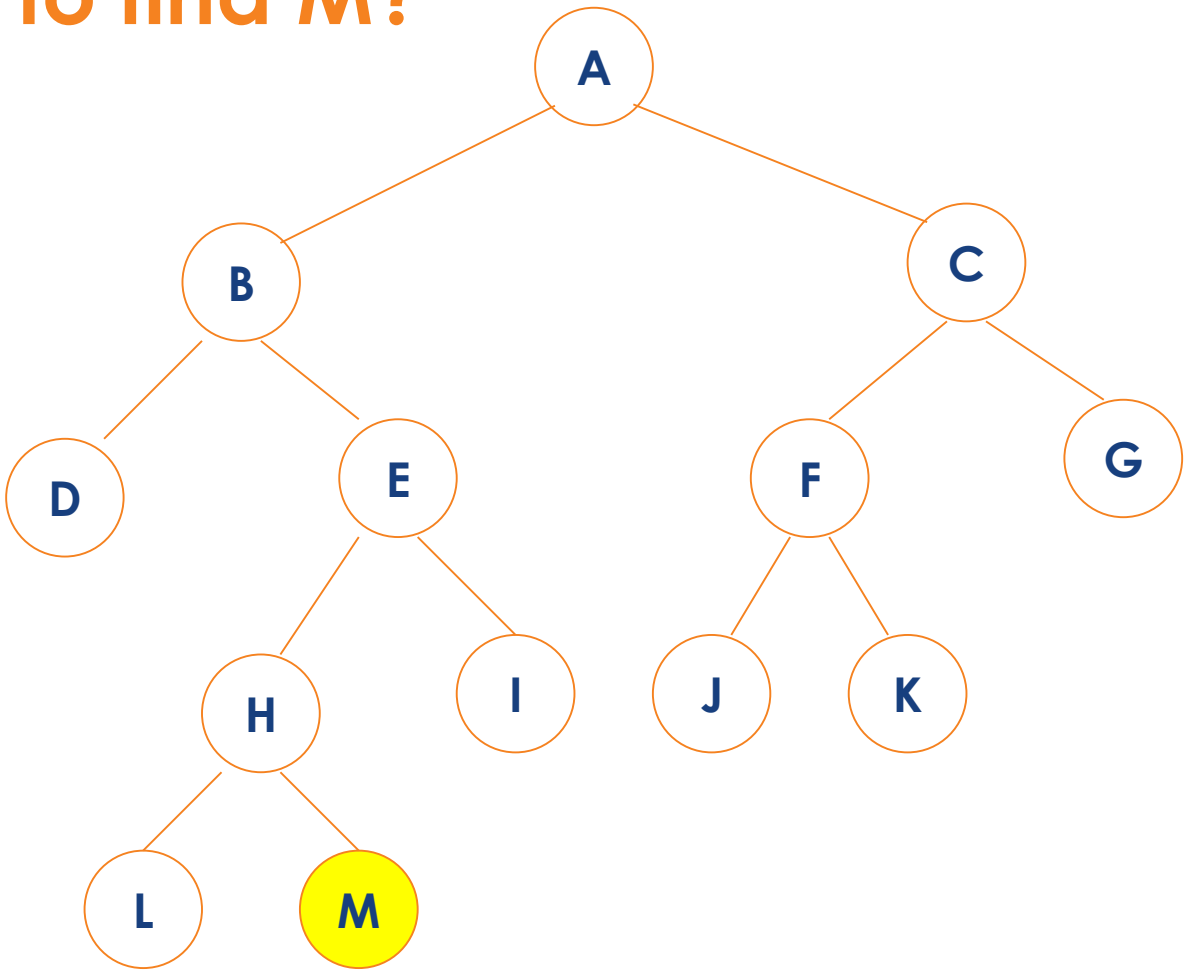
(4.1) If **n** is a goal node, then exit and signal success

(4.2) Otherwise, add the children of **n** to the **end** of **N** then go to step **(2)**

1.2 UNINFORMED SEARCH TECHNIQUES

Breadth First Search (BFS)

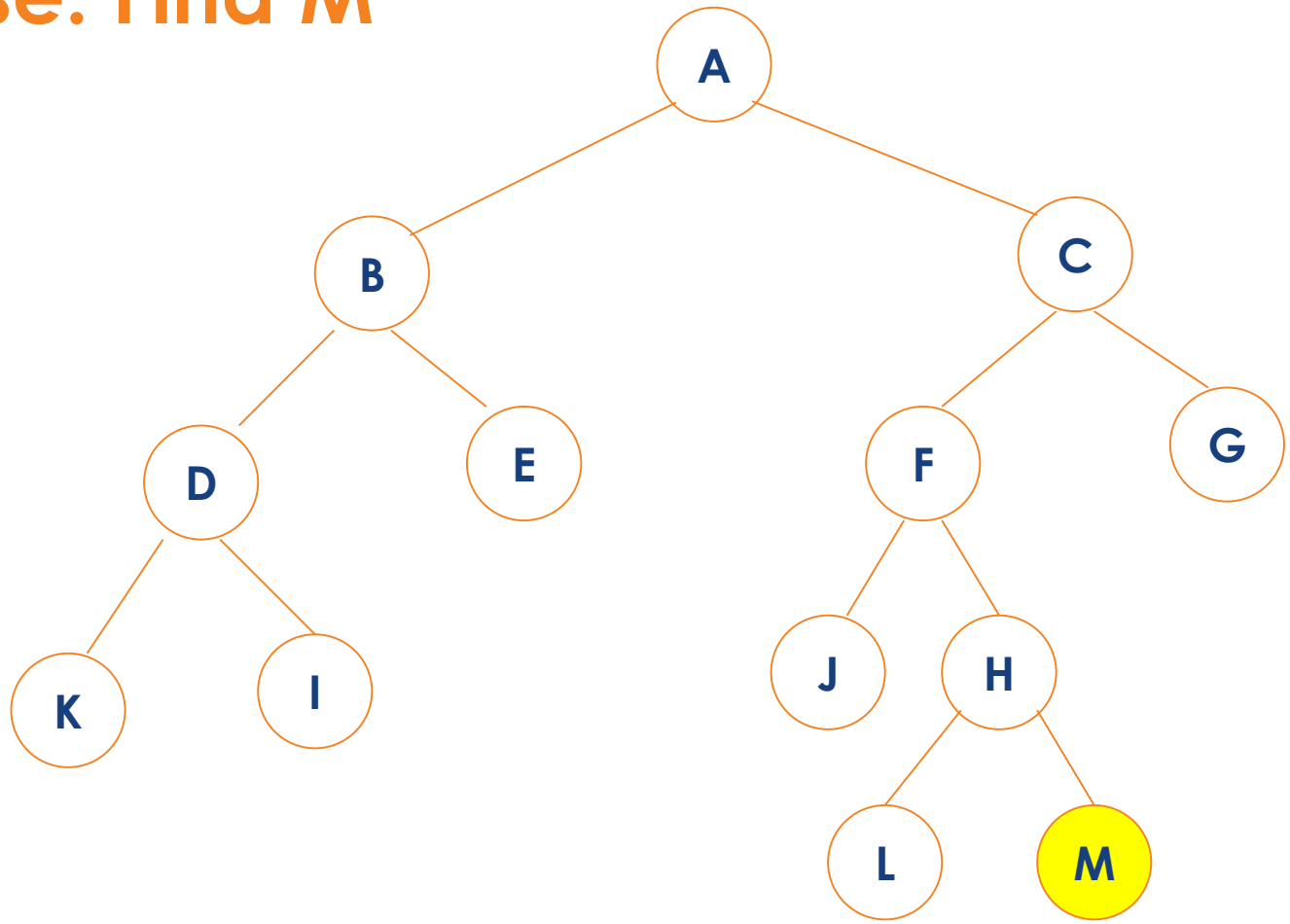
- Visit order to find M?



1.2 UNINFORMED SEARCH TECHNIQUES

Breadth First Search (BFS)

- **Exercise: Find M**



1.2 UNINFORMED SEARCH TECHNIQUES

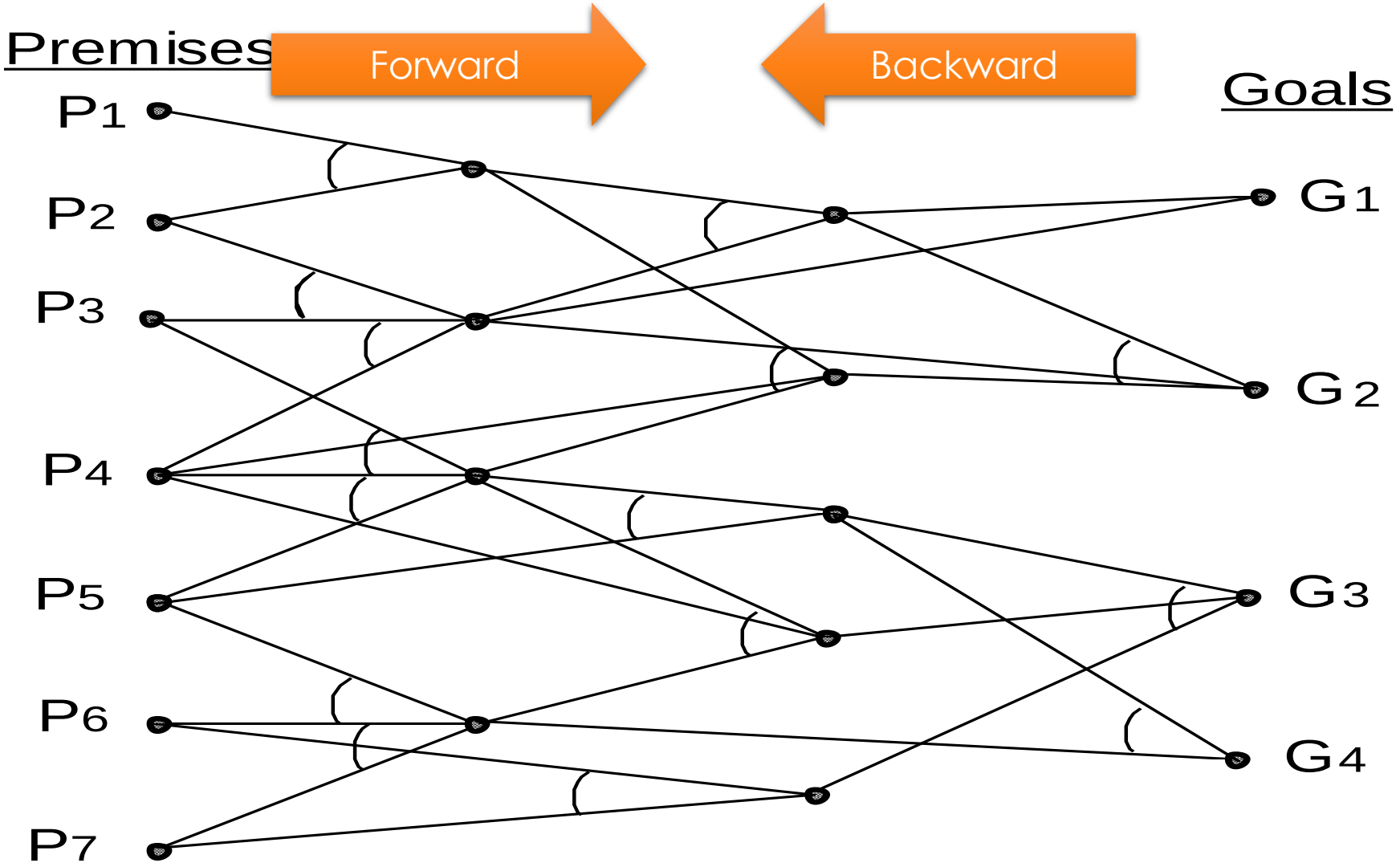
Breadth First Search (BFS)

BFS Drawbacks

- Can consume more memory or storage spaces, due to remembering all past explored solutions/nodes.

1.2 UNINFORMED SEARCH TECHNIQUES

Forward Chaining (BFS) vs. Backward Chaining (DFS)



1.3 WORKSHOP

SEARCH REPRESENTATION

1.3 WORKSHOP SEARCH REPRESENTATION

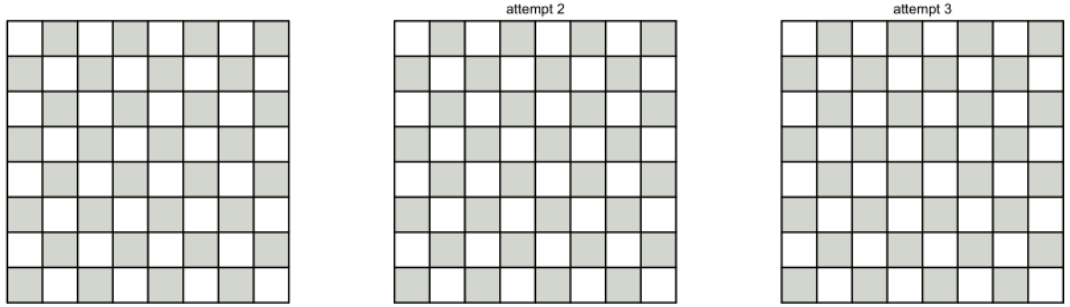
- **Search Modelling & Representation**
  - Pen & Paper Planning
  - Robot Navigation
  - Vehicle Route Planning (VRP)
- **KIE OptaPlanner Tutorial**
  - Optimizing Vehicle Route Planning (VRP)
  - Optimizing Europe Travelling Sales Person (TSP)

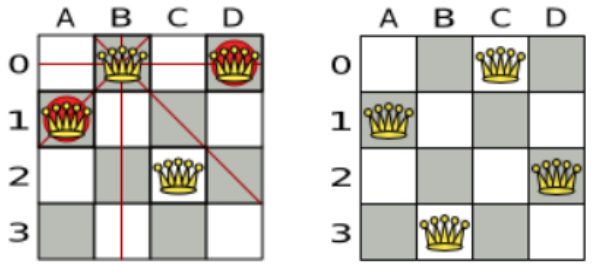
1.3 WORKSHOP SEARCH REPRESENTATION

Search Modelling & Representation

- **Pen & Paper Planning**

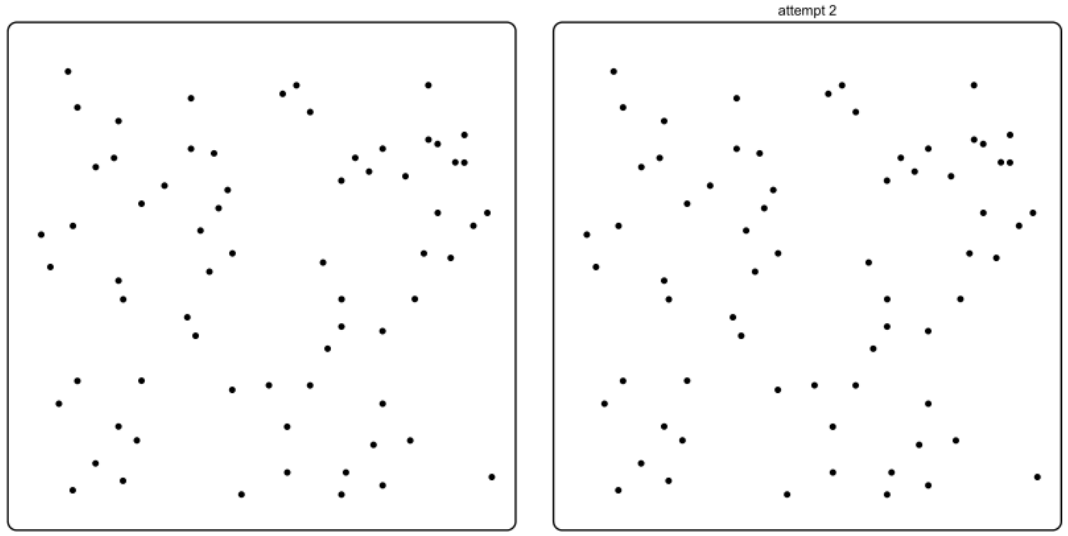
1) Place 8 queens on this chessboard so no 2 queens can attack each other.

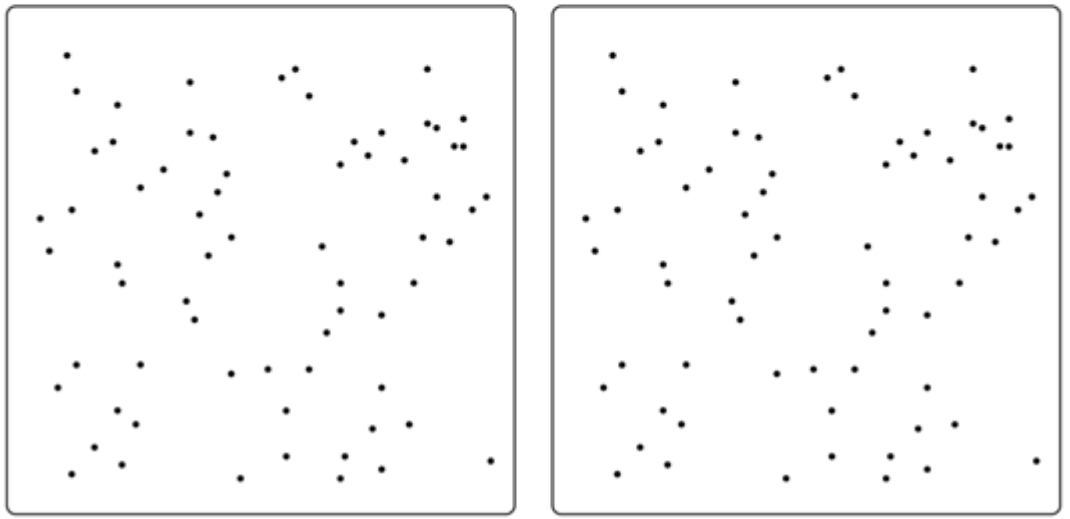




Bad Good

2) Draw the shortest line that connects all dots and returns to its origin.

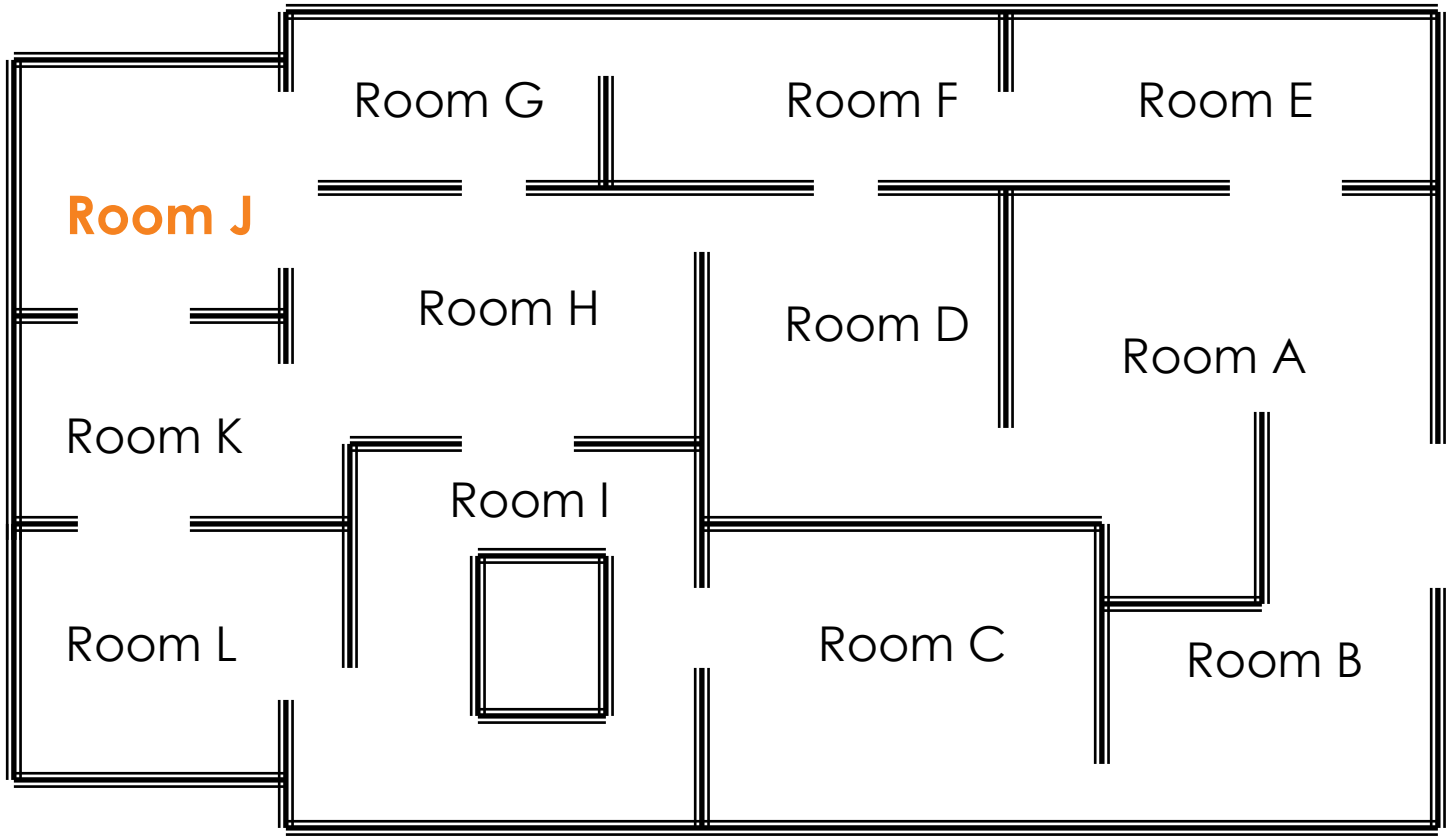


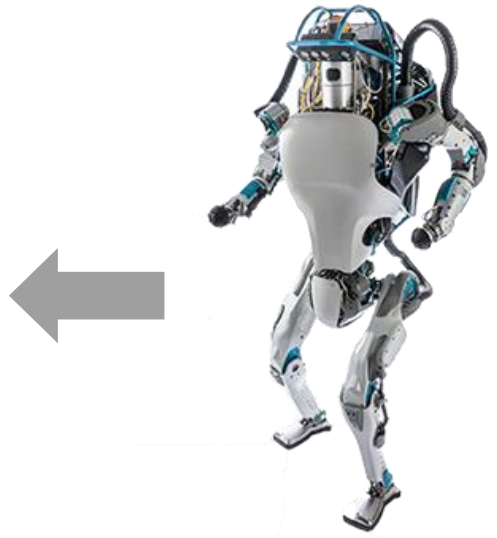


1.3 WORKSHOP SEARCH REPRESENTATION

Search Modelling & Representation

- **Robotics: How to rapidly navigate to Room J ?**



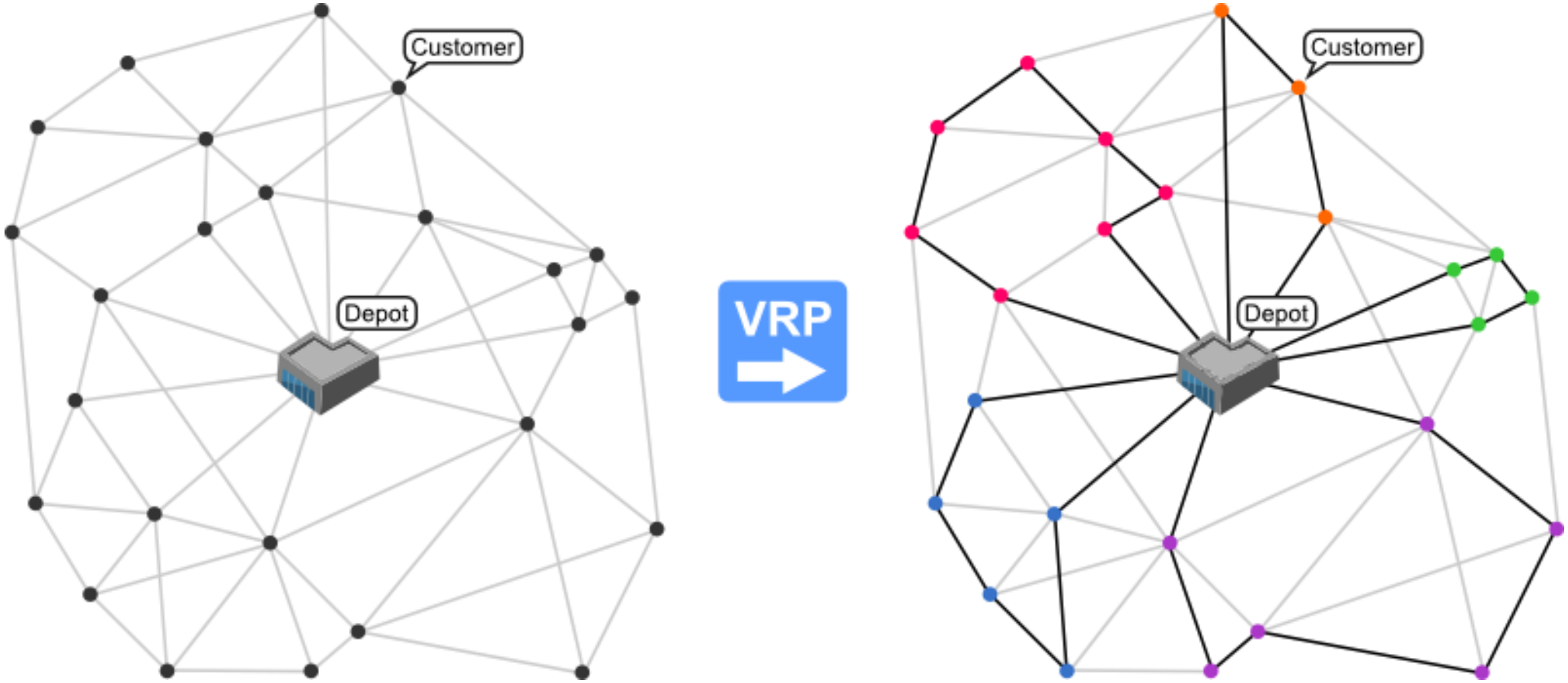


https://static1.squarespace.com/static/57c8a68a20099ef23fb19e90/t/5a32a9804192022be97f9d10/1513269668629/Atlas.png

1.3 WORKSHOP SEARCH REPRESENTATION

Search Modelling & Representation

Vehicle Route Problem



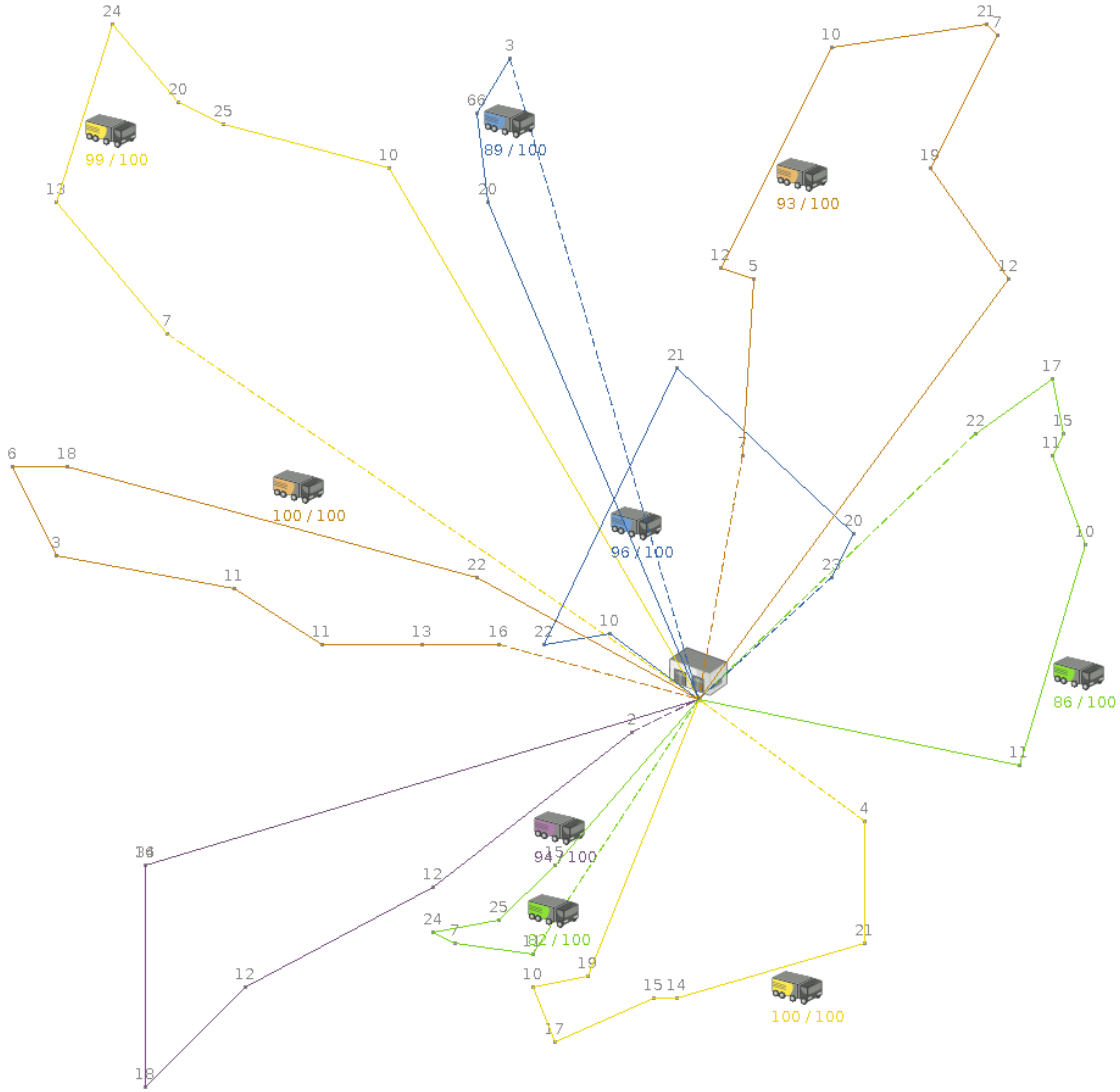
http://neo.lcc.uma.es/vrp/vehicle-routing-problem/

1.3 WORKSHOP SEARCH REPRESENTATION

Search Modelling & Representation

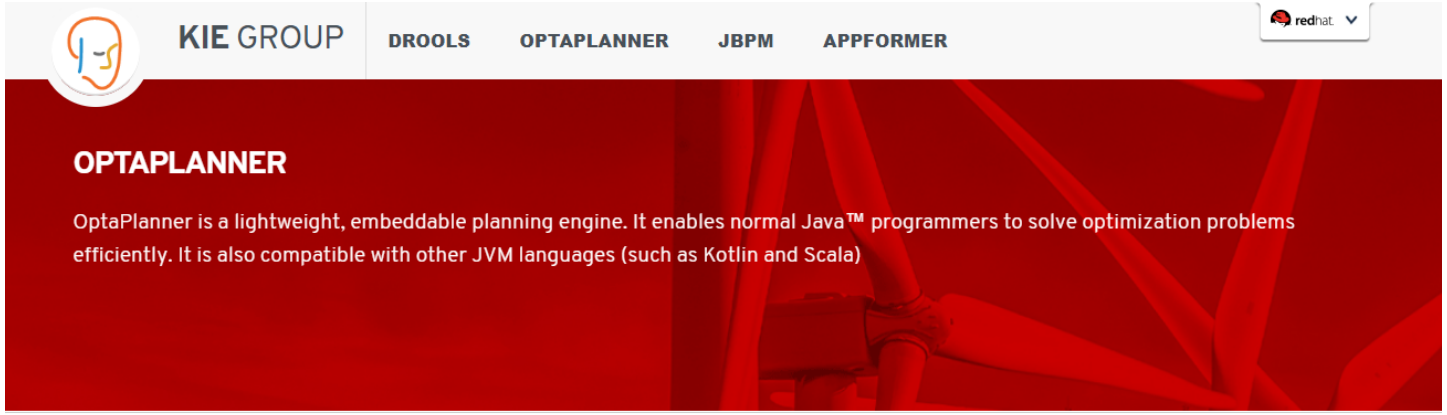
Optimizing Vehicle Route Planning

- We are a logistic company owning a warehouse and 9 delivery trucks. This morning we received 54 customer orders, with different load demand, and different locations. Our truck's maximum load capacity is 100 TVs.
- We want to delivery all customer orders using fewer gasoline. Hence, we'd like to have shortest distance of combined truck delivery routes.



1.3 WORKSHOP SEARCH REPRESENTATION

KIE OptaPlanner Tutorial



JBoss KIE

http://www.kiegroup.org/

DROOLS

Drools is a business rule management system with a forward-chaining and backward-chaining inference based rules engine, allowing fast and reliable evaluation of business rules and complex event processing.

Read more →

OPTAPLANNER

OptaPlanner is a constraint solver that optimizes use cases such as employee rostering, vehicle routing, task assignment and cloud optimization.

Read more →

JBoss KIE OptaPlanner

http://www.optaplanner.org/

JBPM

jBPM is a flexible Business Process Management suite allowing you to model your business goals by describing the steps that need to be executed to achieve those goals.

Read more →

APPFORMER

AppFormer is a low code platform to develop modern applications. It's a powerful tool for developers that can easily build applications by mashing up components and connect them to other Red Hat modules and software.

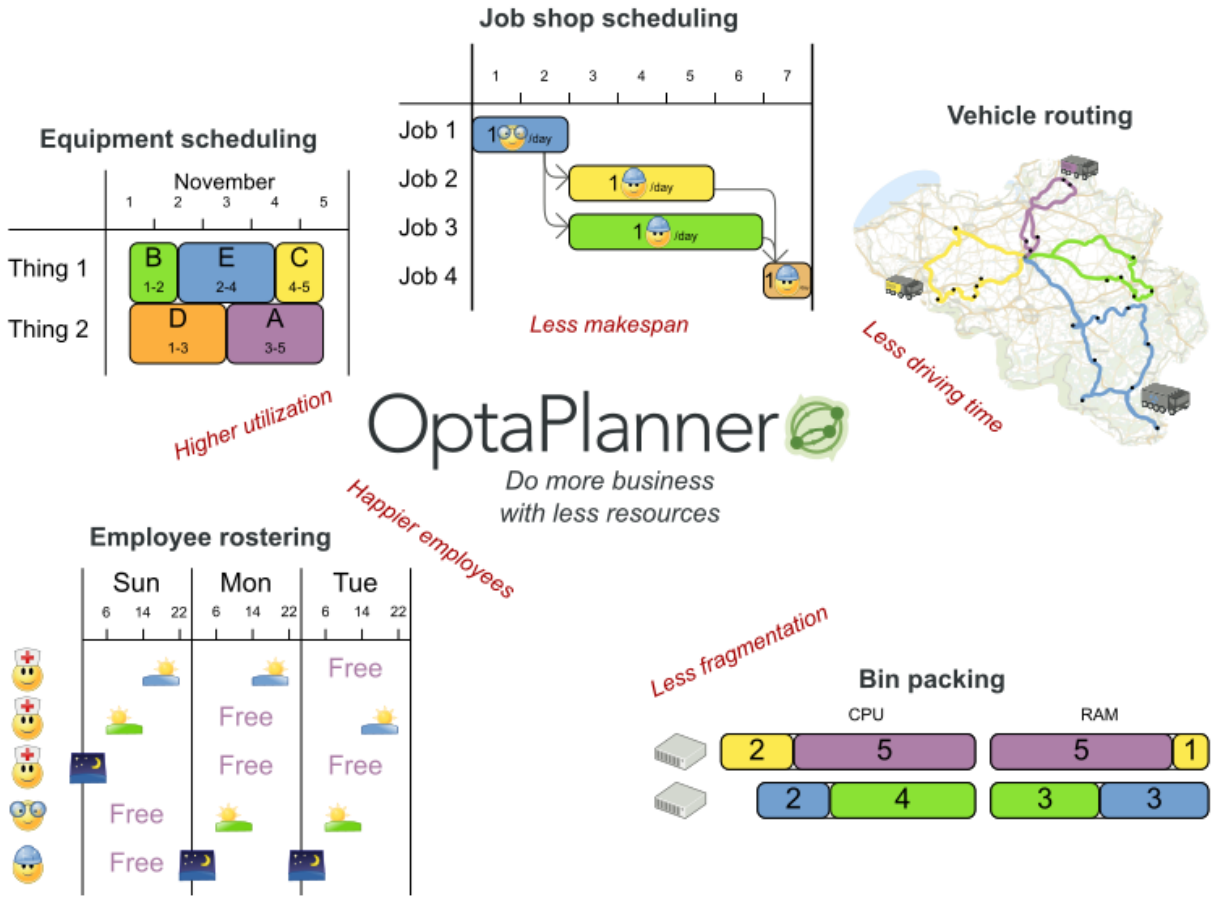
We make building apps looks easy.

Read more →

1.3 WORKSHOP SEARCH REPRESENTATION

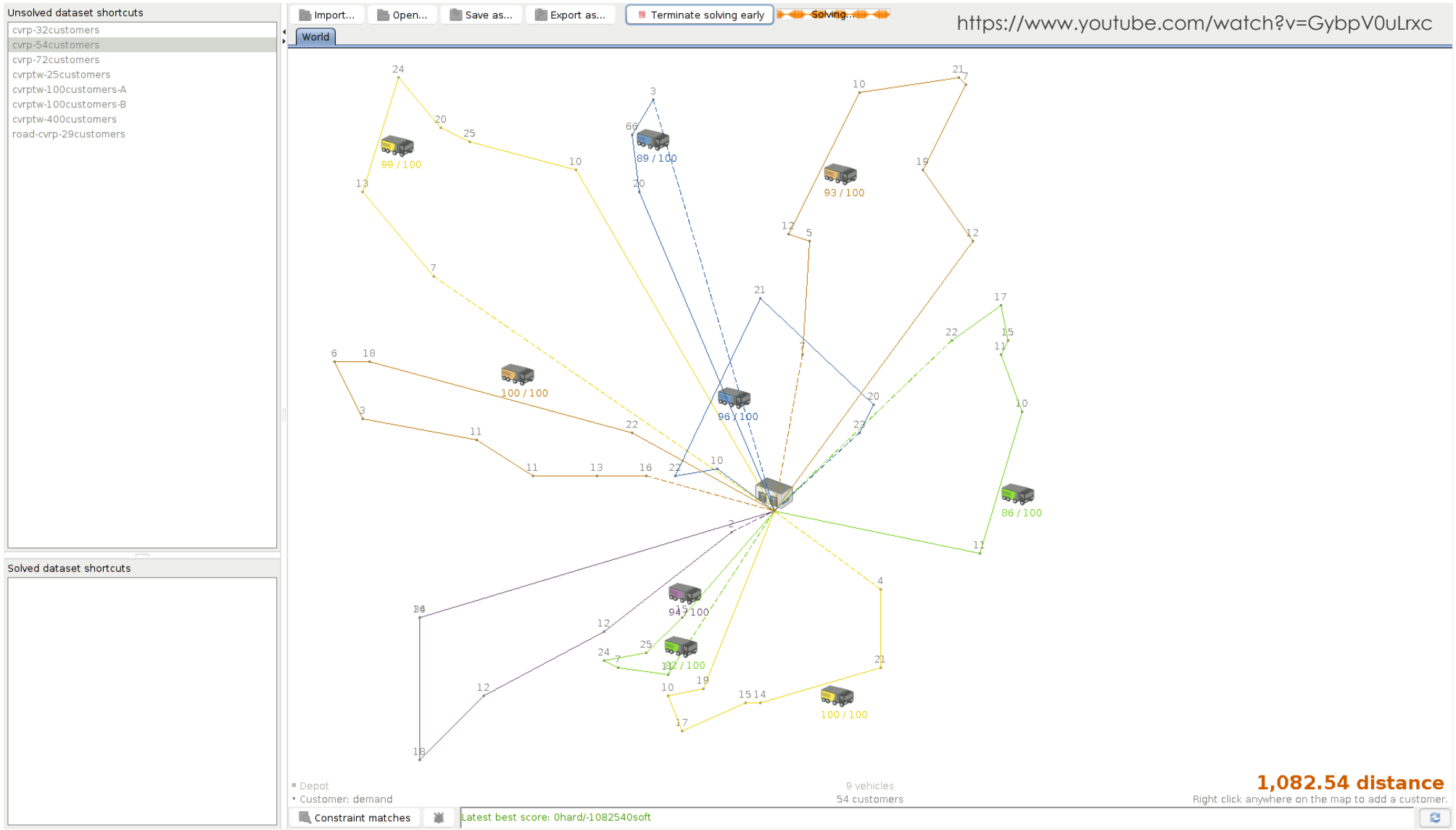
KIE OptaPlanner Tutorial

- **Constrain Satisfaction: Business Resource Optimizer**



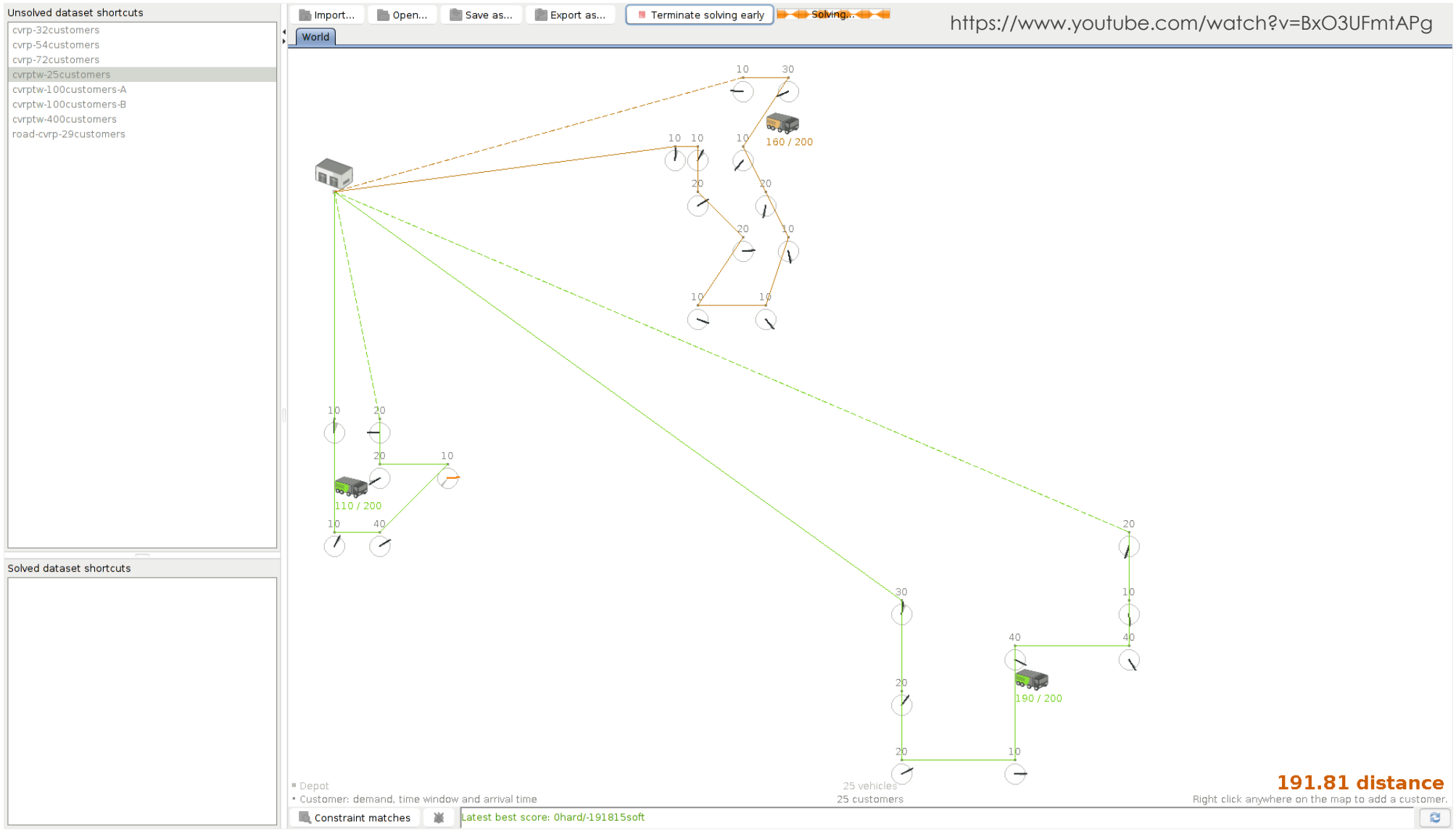
1.3 WORKSHOP SEARCH REPRESENTATION

KIE OptaPlanner Tutorial – VRP: Customer demand (vehicle load)



1.3 WORKSHOP SEARCH REPRESENTATION

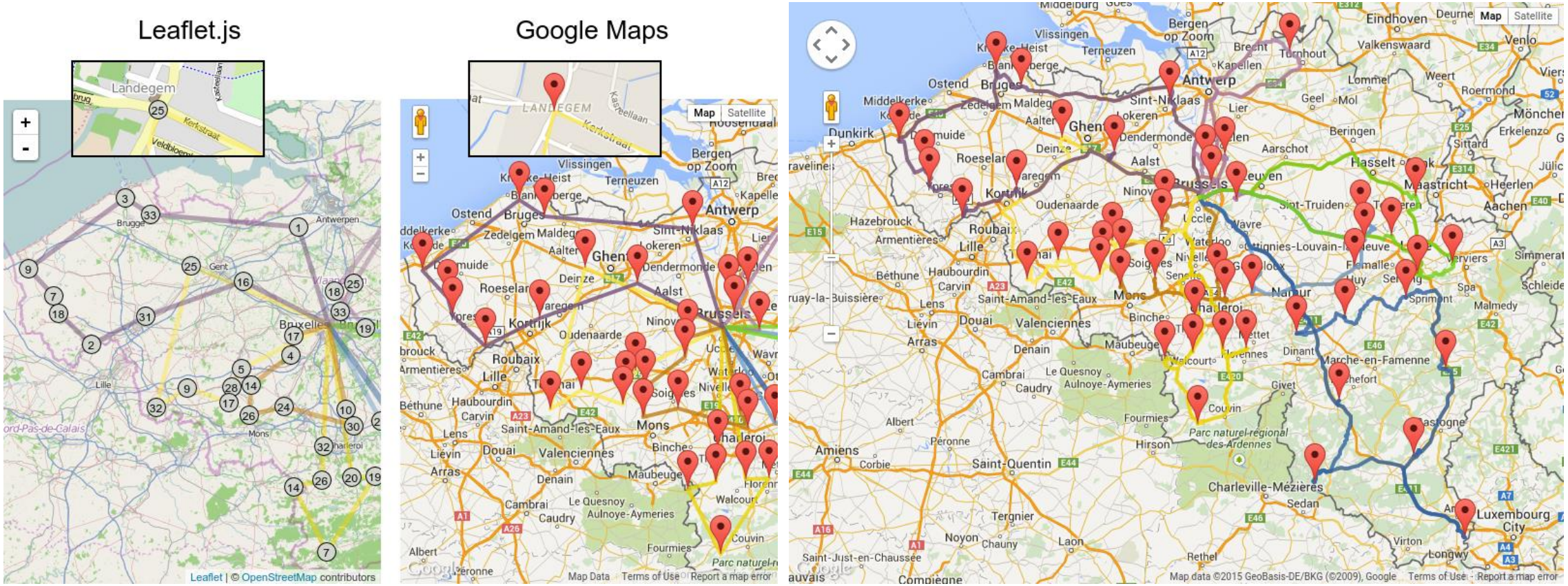
KIE OptaPlanner Tutorial – VRP: Customer demand, Time window



1.3 WORKSHOP SEARCH REPRESENTATION

KIE OptaPlanner Tutorial – VRP with map integration

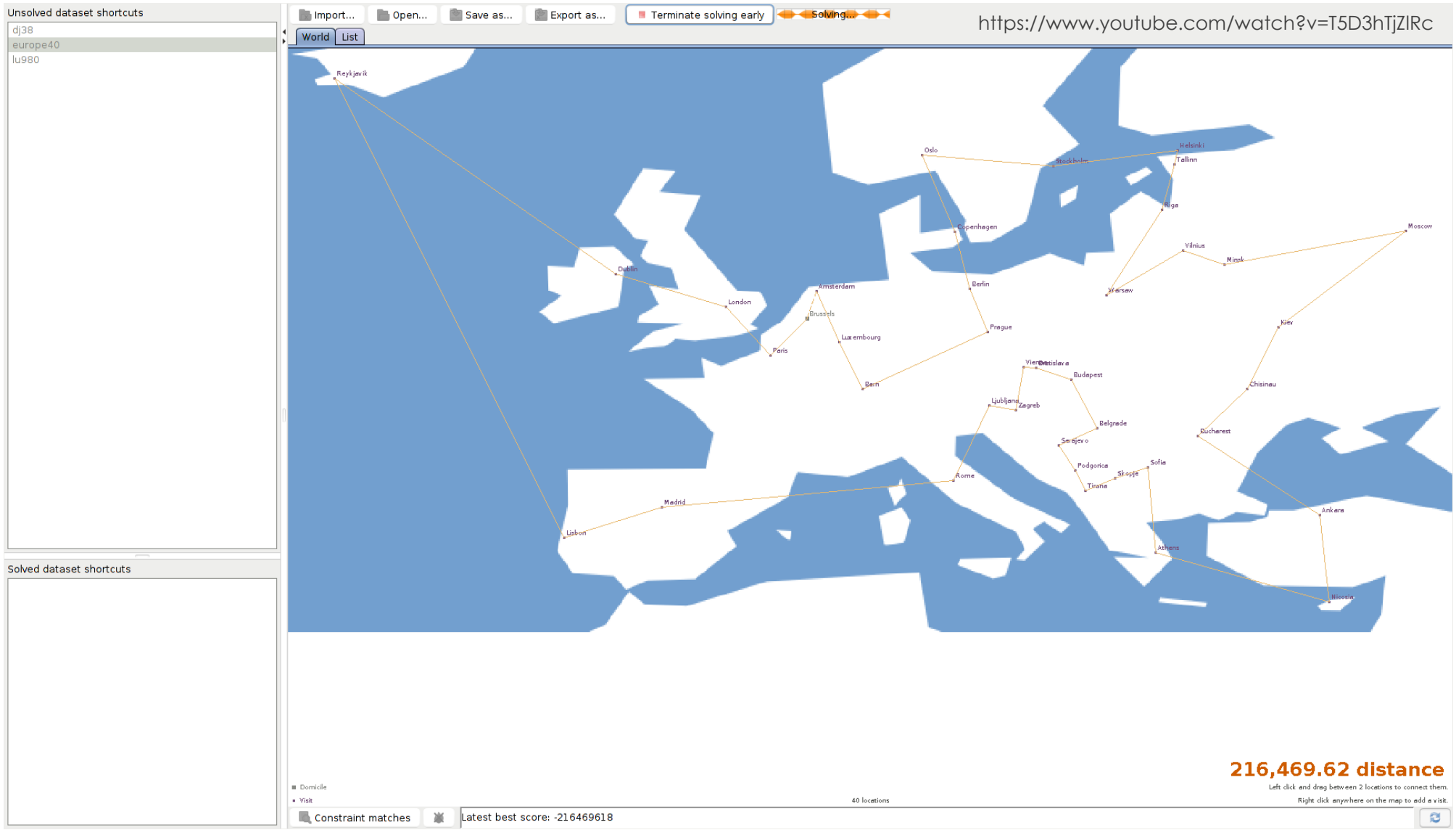
Visualizing Vehicle Routing with Leaflet and Google Maps



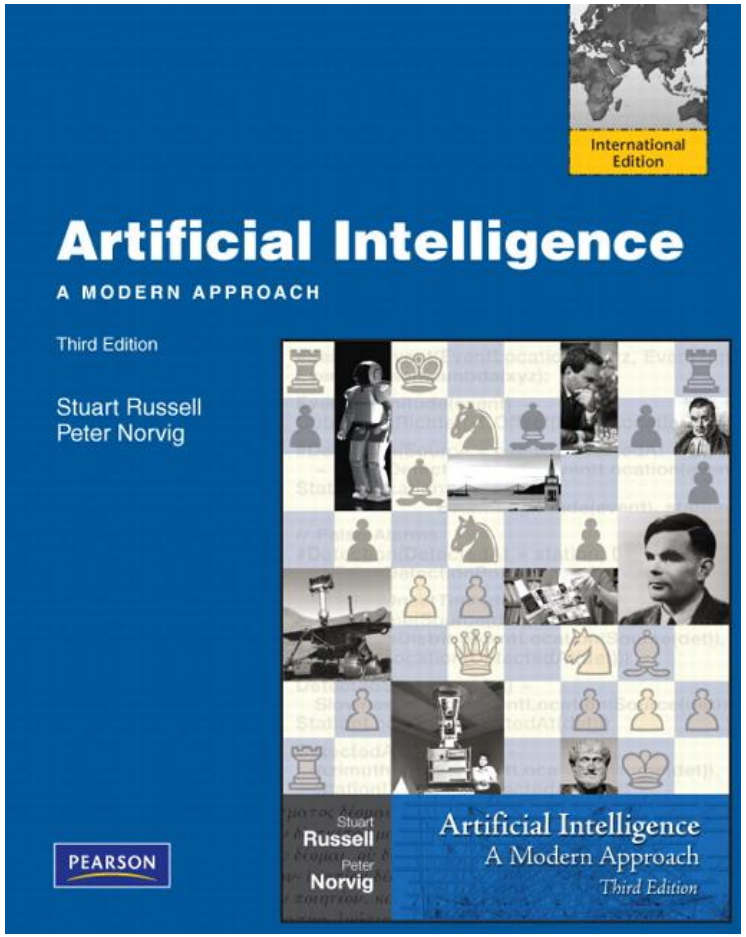
https://www.optaplanner.org/blog/2015/03/10/VisualizingVehicleRoutingWithLeafletAndGoogleMaps.html

1.3 WORKSHOP SEARCH REPRESENTATION

KIE OptaPlanner Tutorial – TSP: Europe cities



DAY 1 REFERENCE



1. **OptaPlanner : Do more business with less recourses**

   http://www.optaplanner.org/learn/slides/optaplanner-presentation/index.html#/1

2. OptaPlanner

   https://www.optaplanner.org/

3. OptaPlanner Use Cases & Demo Videos

   http://www.optaplanner.org/learn/useCases/index.html

4. OptaPlanner Video Tutorials

   http://www.optaplanner.org/learn/video.html

   https://www.youtube.com/user/ge0ffrey2

5. Onne Beek. (2011). Efficient Local Search Methods For Vehicle Routing

   https://lib.ugent.be/fulltxt/RUG01/001/788/544/RUG01-001788544_2012_0001_AC.pdf