# Artificial Intelligence Foundation – JC3001

## Lecture 13: Constraint Satisfaction Problems I

**Prof. Aladdin Ayesh** (aladdin.ayesh@abdn.ac.uk)

**Dr. Binod Bhattarai** (binod.bhattarai@abdn.ac.uk)

**Dr. Gideon Ogunniye,** (g.ogunniye@abdn.ac.uk)

September 2025

Material adapted from:

Russell and Norvig (AIMA Book): Chapter 6

# Course Progression

- Part 1: Introduction
  1. Introduction to AI ✓
  2. Agents ✓
- Part 2: Problem-solving
  1. Search 1: Uninformed Search ✓
  2. Search 2: Heuristic Search ✓
  3. Search 3: Local Search ✓
  4. Search 4: Adversarial Search ✓
- Part 3: Reasoning and Uncertainty
  1. **Reasoning 1: Constraint Satisfaction**
  2. Reasoning 2: Logic and Inference
  3. Probabilistic Reasoning 1: BNs
  4. Probabilistic Reasoning 2: HMMs
- Part 4: Planning
  1. Planning 1: Intro and Formalism
  2. Planning 2: Algos and Heuristics
  3. Planning 3: Hierarchical Planning
  4. Planning 4: Stochastic Planning
- Part 5: Learning
  1. Learning 1: Intro to ML
  2. Learning 2: Regression
  3. Learning 3: Neural Networks
  4. Learning 4: Reinforcement Learning
- Part 6: Conclusion
  1. Ethical Issues in AI
  2. Conclusions and Discussion

# Objectives

- Defining Constraint Satisfaction Problems (CSP)
- CSP examples
- Backtracking search for CSPs
- Local search for CSPs
- Problem structure and problem decomposition

- In many optimization problems, the **path** to a goal is irrelevant
  - the goal state itself is the solution
- State space = a set of goal states
  - find one that satisfies constraints (e.g., no two classes at same time)
  - or find **optimal** one (e.g., highest possible value, least possible cost)
- Iterative improvement algorithms – keep a single “current“ state, try to improve it
  - Constant space
  - Suitable for both offline and online search

- In many optimization problems, the **path** to a goal is irrelevant
  - the goal state itself is the solution
- State space = a set of goal states
  - find one that satisfies constraints (e.g., no two classes at same time)
  - or find **optimal** one (e.g., highest possible value, least possible cost)
- Iterative improvement algorithms – keep a single "current" state, try to improve it
  - Constant space
  - Suitable for both offline and online search
- Local search algorithms: Hill Climbing, Simulated Annealing, GA
  - Consider states monolithic data structures
  - Search ignores internal structure almost completely
  - Genetic algorithms do not <u>reason</u> about internal structure

- Constraint Satisfaction Problems

# Defining Constraint Satisfaction Problems

A constraint satisfaction problem (CSP) consists of three components, $X$, $D$, and $C$:

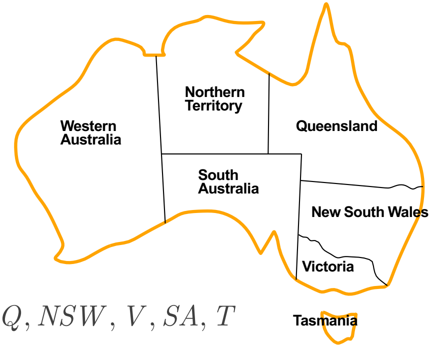
- $X$ is a set of variables, $\{X_1, \ldots, X_n\}$.
- $D$ is a set of domains, $\{D_1, \ldots, D_n\}$, one for each variable
- $C$ is a set of constraints that specify allowable combination of values

CSPs deal with assignments of values to variables.

- A complete assignment is one in which every variable is assigned a value
  - A solution to a CSP is a consistent, complete assignment.
- A partial assignment is one that leaves some variables unassigned.
- Partial solution is a partial assignment that is consistent

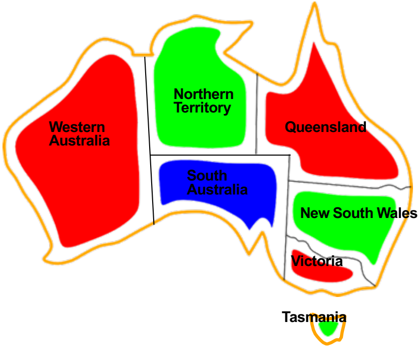
# Constraint satisfaction problems (CSPs)

- Standard search problem:
  - **state** is a “black box” — any old data structure that supports goal test, eval, successor
- CSP:
  - **state** is defined by *variables* $X_i$ with values from *domain* $D_i$
  - **goal test** is a set of constraints specifying allowable combinations of values for subsets of variables
- Simple example of a **formal representation language**
- Allows useful **general-purpose** algorithms with more power than standard search algorithms



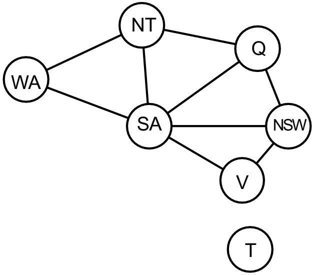

- Variables: $WA, NT, Q, NSW, V, SA, T$
- Domains: $D_i = red, green, blue$
- Constraints: adjacent regions must have different colours
  e.g., $WA \neq NT$ (if the language allows this), or
  $(WA, NT) \in \{(red, green), (red, blue), (green, red), (green, blue), \dots\}$

# Example: Map-Colouring contd.




Solutions are assignments satisfying all constraints, e.g.,
$\{WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green\}$

- **Binary CSP:** each constraint relates at most two variables
- **Constraint graph:** nodes are variables, arcs show constraints



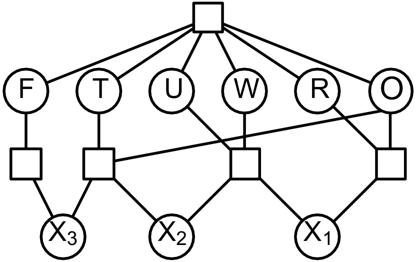

General-purpose CSP algorithms use the graph structure
to speed up search. E.g., Tasmania is an independent subproblem!

- Discrete variables
  - finite domains; size $d \Rightarrow O(d^n)$ complete assignments
    e.g., Boolean CSPs, incl. Boolean satisfiability (NP-complete)
    infinite domains (integers, strings, etc.)
    e.g., job scheduling, variables are start/end days for each job
  - need a constraint language, e.g., $StartJob_1 + 5 \leq StartJob_3$
  - linear constraints solvable, nonlinear undecidable
- Continuous variables
  e.g., start/end times for Hubble Telescope observations
  - linear constraints solvable in poly time by LP methods

# Varieties of constraints

- **Unary** constraints involve a single variable,
  e.g., $SA \neq green$
- **Binary** constraints involve pairs of variables,
  e.g., $SA \neq WA$
- **Higher-order** constraints involve 3 or more variables,
  e.g., cryptarithmetic column constraints
- Preferences (soft constraints),
  e.g., $red$ is better than $green$
- often representable by a cost for each variable assignment
  $\rightarrow$ constrained optimization problems

$$\begin{array}{cccc} & T & W & O \\ + & T & W & O \\ \hline F & O & U & R \end{array}$$




- **Variables:** $F, T, U, W, R, O, X_1, X_2, X_3$
- **Domains:** $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- **Constraints**
  $\mathit{AllDiff}\,(F, T, U, W, R, O)$
  $O + O = R + 10 \cdot X_1$, etc.

# Real-world CSPs

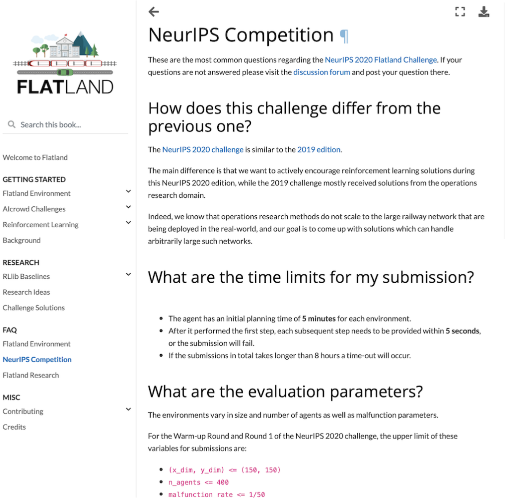
- Assignment problems
  e.g., who teaches what class
- Timetabling problems
  e.g., which class is offered when and where?
- Hardware configuration
- Spreadsheets
- Transportation scheduling
- Factory scheduling
- Floorplanning

Notice that many real-world problems involve real-valued variables

# A Story about ML vs. GOFAI

## Flatland Challenge

- Scheduling for a large rail network in Switzerland
  - Traditionally solved by operations research
  - Challenge in scaling up to realistic networks
- Posted into NeurIPS to attract ML solutions



FLATLAND

Search this book...

Welcome to Flatland

GETTING STARTED

Flatland Environment

AIcrowd Challenges

Reinforcement Learning

Background

RESEARCH

RLlib Baselines

Research Ideas

Challenge Solutions

FAQ

Flatland Environment

NeurIPS Competition

Flatland Research

MISC

Contributing

Credits

# NeurIPS Competition ¶

These are the most common questions regarding the NeurIPS 2020 Flatland Challenge. If your questions are not answered please visit the discussion forum and post your question there.

## How does this challenge differ from the previous one?

The NeurIPS 2020 challenge is similar to the 2019 edition.

The main difference is that we want to actively encourage reinforcement learning solutions during this NeurIPS 2020 edition, while the 2019 challenge mostly received solutions from the operations research domain.

Indeed, we know that operations research methods do not scale to the large railway network that are being deployed in the real-world, and our goal is to come up with solutions which can handle arbitrarily large such networks.

## What are the time limits for my submission?

- The agent has an initial planning time of **5 minutes** for each environment.
- After it performed the first step, each subsequent step needs to be provided within **5 seconds**, or the submission will fail.
- If the submissions in total takes longer than 8 hours a time-out will occur.

## What are the evaluation parameters?

The environments vary in size and number of agents as well as malfunction parameters.

For the Warm-up Round and Round 1 of the NeurIPS 2020 challenge, the upper limit of these variables for submissions are:

- `(x_dim, y_dim) <= (150, 150)`
- `n_agents <= 400`
- `malfunction rate <= 1/50`

# Flatland Results



- Machine Learning approaches have shown to solve the problem

**NeurIPS 2020 Flatland Winners**

Flatland announcement

**Round 2 has finished, concluding the NeurIPS 2020 edition!**

We have received over 2'000 submissions from over 700 participants coming from over 51 countries! We have been thrilled by your contributions and enthusiasm.

And the winners are… 🥁

**RL Track**

- 🥇 **JBR_HSE**, 214.15 points
- 🥈 **ai-team-flatland**, 181.497 points
- 🥉 **SSG-RL**, 143.985 points

**Non-RL Track**

- 🥇 **An_Old_Driver**, 297.507 points
- 🥈 **MasterFlatland**, 296.347 points
- 🥉 **vetrov_andrew**, 273.339 points

Congratulations to all of them! The race was intense 🎉 👏

https://discourse.aicrowd.com/t/neurips-2020-flatland-winners/4010

- Machine Learning approaches have shown to solve the problem

- Except they **lost**, bad to GOFAI

**NeurIPS 2020 Flatland Winners**

Flatland announcement

**Round 2 has finished, concluding the NeurIPS 2020 edition!**

We have received over 2'000 submissions from over 700 participants coming from over 51 countries!
We have been thrilled by your contributions and enthusiasm.

And the winners are… 🥁

**RL Track**

- 🥇 **JBR_HSE**, 214.15 points
- 🥈 **ai-team-flatland**, 181.497 points
- 🥉 **SSG-RL**, 143.985 points

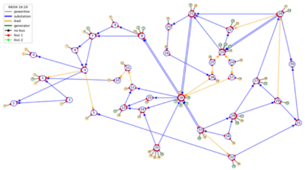
**Non-RL Track**

- 🥇 **An_Old_Driver**, 297.507 points
- 🥈 **MasterFlatland**, 296.347 points
- 🥉 **vetrov_andrew**, 273.339 points

Congratulations to all of them! The race was intense 🎉 👏

https://discourse.aicrowd.com/t/neurips-2020-flatland-winners/4010

- Same story



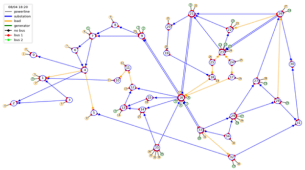
### Learning to Run a Power Network with Trust

LIVE

- Winner (New Entrant Track) 4,500 USD
- Winner (Anyone Track) 7,500 USD
- Best Student Prize 1,000 USD

On the way towards a sustainable future and following up the success of L2RPN 2020 NeurIPS competition, this competition aims at unleashing the power of artificial intelligence even further for our real-world industrial application: controlling electricity power transmission in real-time and moving closer to truly “smart” grids using underutilized flexibilities. In 2020, participants were asked to develop an agent to be robust to unexpected events and keep delivering reliable electricity everywhere even in difficult circumstances.

- Same story
- 3 out of 4 competitions won by GOFAI (Using CSP-style algorithms)
  - https://www.youtube.com/watch?v=q4F-peykOFE
  - https://www.youtube.com/watch?v=8gSCxHZY_1M
  - https://www.youtube.com/watch?v=SL-iBAQTJKM



**Learning to Run a Power Network with Trust**

LIVE

- Winner (New Entrant Track) 4,500 USD
- Winner (Anyone Track) 7,500 USD
- Best Student Prize 1,000 USD

On the way towards a sustainable future and following up the success of L2RPN 2020 NeurIPS competition, this competition aims at unleashing the power of artificial intelligence even further for our real-world industrial application: controlling electricity power transmission in real-time and moving closer to truly "smart" grids using underutilized flexibilities. In 2020, participants were asked to develop an agent to be robust to unexpected events and keep delivering reliable electricity everywhere even in difficult circumstances.

To continue in the next session.