



# Artificial Intelligence Methods (G52AIM)

---

Dr Rong Qu

[rxq@cs.nott.ac.uk](mailto:rxq@cs.nott.ac.uk)

***Constructive Heuristic Methods***



# Constructive Heuristics – method

---

- Start from an “empty solution”
- Repeatedly, extend the current solution until a complete solution is constructed
- Use heuristics to try to extend in such a way that the final solution is a good one

Some of the slides in this part are based on Dr Parkes’ previous teaching materials. See “Constructive Heuristic Methods” at <http://www.cs.nott.ac.uk/~ajp/courses/g52aim/>.



# Constructive Heuristics – method

---

It is essential to know the difference between:

- Constructive methods
  - Extend empty solution until get complete solution
- Local search
  - Take complete solution and try to improve it via local moves



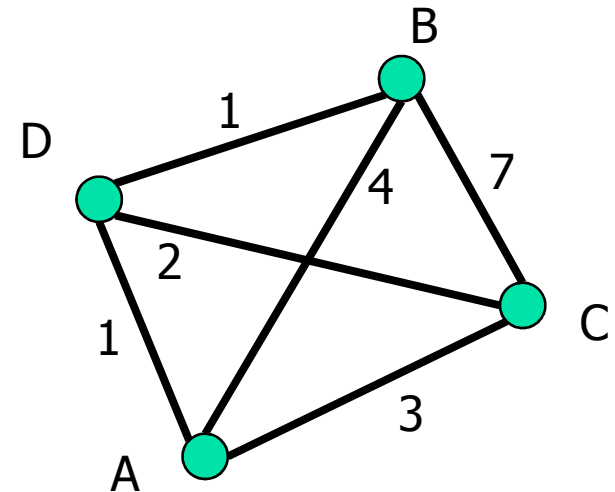
# Constructive Heuristics for TSP

---

- Start: Pick a start city but an empty path
- Repeatedly:
  - Extend the path by adding an edge to an unvisited city selected using a heuristic choice
- Until no more unvisited cities, then close the path to give a tour

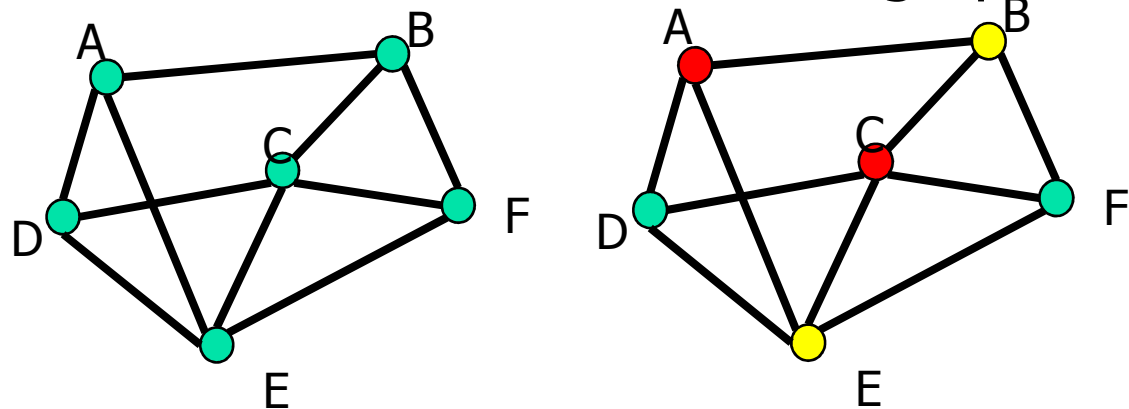
# Constructive Heuristics for TSP

- Heuristics
  - Nearest Neighbour (NN) Heuristic
    - Pick the next city which is the nearest unvisited city
  - Any other good heuristics?
    - Online demo
  - NN from A
    - A-D-B-C-A = 1+1+7+3=12
    - A-C-D-B-A = 3+2+1+4 = 10
      - Optimal?



# Constructive Heuristics for GC

- Graph colouring
  - Given a graph  $G(V, E)$ , assign colours to vertices
  - Constraint
    - Adjacent vertices cannot be coloured the same
  - Objective
    - Minimize: the number of colours to the graph





# Constructive Heuristics for GC

---

- Heuristics
  - Largest degree
    - Pick the vertices with the largest degree
    - Most Constraining
  - Saturation degree
    - Pick vertices with the fewest remaining colours
    - Most Constrained
  - Any other good heuristics?



# Constructive Heuristics

---

- Putting this into the context of depth first search in G51IAI\*
  - At each choice suppose that branches are ordered left-to-right by the heuristic, with the preferred option on left
  - Standard one-shot construction is just to take one branch of the search tree

\* You should be familiar with the blind search methods in G51IAI



# Constructive Heuristics

---

- Putting this into the context of depth first search in G51IAI\*
  - To avoid mistakes in picking the “wrong” branches, heuristics can help to pick branches which lead to relatively good solutions

\* You should be familiar with the blind search methods in G51IAI



# Heuristics

---

- Variable selection
  - Which variable to work on
  - In GC: which vertex to colour
- Value selection
  - Which value to assign
  - In GC: assign which colour to the chosen vertex



# Heuristics

---

- Variable selection
  - Most constrained
  - In GC: largest degree
- Value selection
  - Least constraining: imposes fewest constraints on remainder of problem
  - In GC: which colour leaves the most colours to the remaining vertices



# Constructive Heuristics

---

- Generally give better answers than random methods
- Very quick, but usually far from optimal
- Widely used with other methods
- Often used as initialization for meta-heuristics
  - Pick the best solution from several runs



# Constructive Heuristics - hybridize

---

- As initialization for local search
  - Pick the best solution from several runs
- Hybridized with other meta-heuristics
  - With local search: GRASP, etc
  - With Genetic algorithms: Memetic algorithms



# Constructive Heuristics - hybridize

---

- GRASP
  - Greedy Randomized Adaptive Search Procedure
  - Hybrid of
    - Constructive methods
      - Randomized & adaptive
    - Local search
      - Pick your favourite



# Constructive Heuristics - hybridize

---

- GRASP

Loop

Create a solution,  $s$ , using randomized constructor  $g$  utilizing RCL

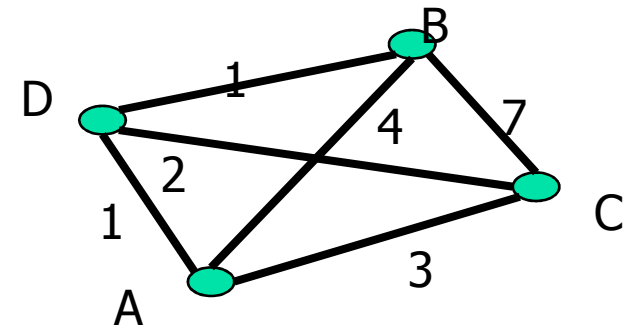
Improve  $s$  using a local search

End loop

# Constructive Heuristics - hybridize

## ■ GRASP

- Main idea of “Restricted candidate list” (RCL)
- At each iteration, within randomized constructor  $g$ 
  1. Use heuristic to select a limited number of good solution components; the RCL
  2. Randomly select a choice from the RCL
  3. Use this in order to extend the current partial assignment





# Constructive Heuristics - hybridize

---

- GRASP

- Main idea of “Restricted candidate list” (RCL)
- Size of RCL  $r$ 
  - If  $r$  is too big, then  $g$  becomes random
  - If  $r$  is 1, then  $g$  becomes pure greedy heuristic
  - Typical size of RCL is 3~5, but maybe problem dependent



# Learning Objectives

---

- Know the general idea of constructive heuristics
- Know GRASP – so you could apply them in your coursework if you had to 😊