Workshop 2: Approximation Algorithms

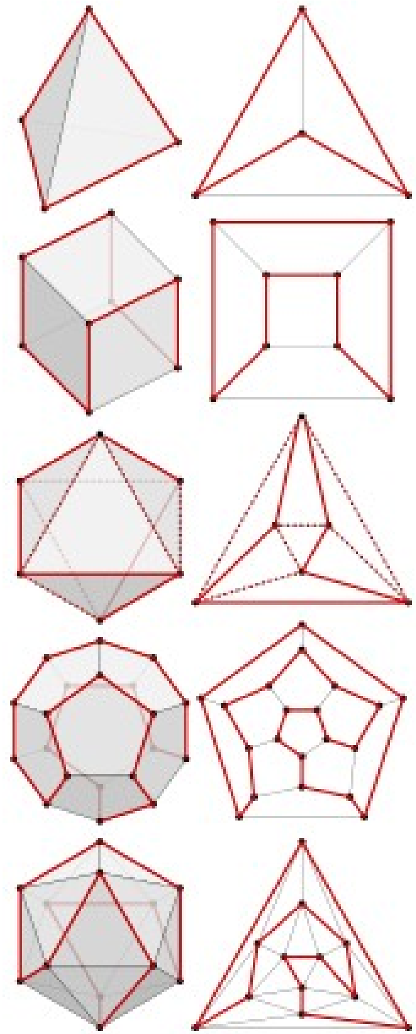
CITS3001 Algorithms, Agents and Artificial Intelligence

Tim French
Dept of Computer Science and Software Engineering
The University of Western Australia

2021, Semester 2

An NP compete problem

- A Hamiltonian Cycle is a path through a graph that starts and ends at the same vertex , and visits every other vertex of the graph exactly once.
- Consider the problem of given a graph, finding a Hamiltonian Cycle, if one exists.
- Is this problem easier or harder than the travelling salesmen problem?
- An algorithm is *correct* if when it returns, it always gives the right answer, and is *compete* if it always returns.
- Describe a correct and complete algorithm for this problem?
- What is the most efficient correct and complete method for solving this problem?
- Describe a correct, but not complete, algorithm to solve this problem.



State space search

- Suppose we have a 0-1 knapsack problem, with *n* items, each with a weight an value, and a knapsack of size *W*.
- How would we express the solution space of this problem?
- How big is the solution space?
- What would a suitable neighbourhood function be?
- What are some search strategies that could be used? Are there any clever optimisations we can use? Are there any breaking cases where our search might fail?

Genetic Algorithms

- A genetic algorithm uses a population of candidate solutions, a machanism for mutating and combining (or breeding new solutions) and the principle of survival of the fittest to evolve a near optimal solution.
- Use these principles to build a genetic algorithm to solve the 0-1 knapsack problem. The key parts are:
  - A *genome*, which is a vector that represents a single solution
  - A method to *mutate* and *crossover* the genome
  - A *fitness function*, and
  - A *selection mechanism* to determine which solutions survive.
- Try implementing this algorithm and compare the results to your algorithm for Lab 2.