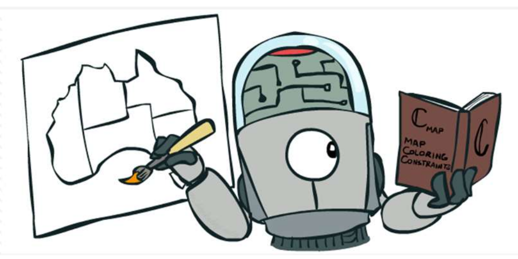
COMPUTATIONAL INTELLIGENCE
CHAPTER: INTRODUCTION ON OPTIMIZATION

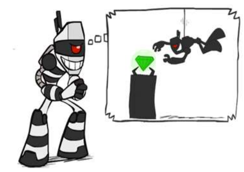
Dr. Siamak Sarmady (UUT)

Constraint Satisfaction Problems



What is Search For?

1. **Planning:** sequences of actions
   - The path to the goal (and its cost) is the important thing
   - Heuristics give problem-specific guidance
     (measure of progress towards goal)



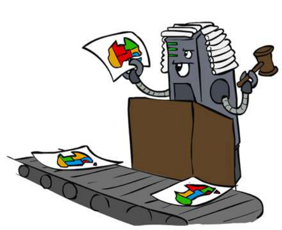
Goal State? How you do that (Planning)?

2. **Identification:** assignments to variables
   - Finding a well formed goal state (solution) is important, not the path (8 queens, Coloring)
   - We want to find assignments to some variables
   - We may use CSP algorithms for identification problems

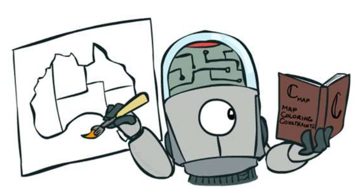
Just wants to know where (in what state) it is
We are not interested in how to reach it

Constraint Satisfaction Problems

- A special subset of search problems…
- **State:** is defined by variables X_i with values from a domain D e.g. X_i: colors for states, D: RGB
- **Domain:** Sometimes D depends on i (and the current state)
- **Goal test:** is a set of constraints rules (e.g. not same colors for neighbor states + all vars. set)



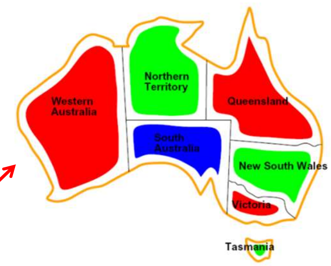
Check different possible states to see whether they are a Goal states or not



Now we just have a series of guides to judge whether a state is goal or not

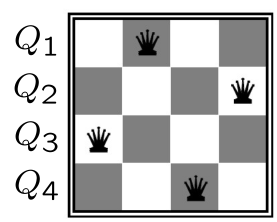
Example: Map Coloring

- **Variables:** WA, NT, Q, NSW, V, SA, T
- **Domains:** D= {red, green, blue}
- **Constraints:** adjacent regions must have different colors
- **Solutions:** are assignments satisfying all constraints
  - {WA=red, NT=green, Q=red, NSW=green, V=red, SA=blue, T=green}



Example: N-Queens

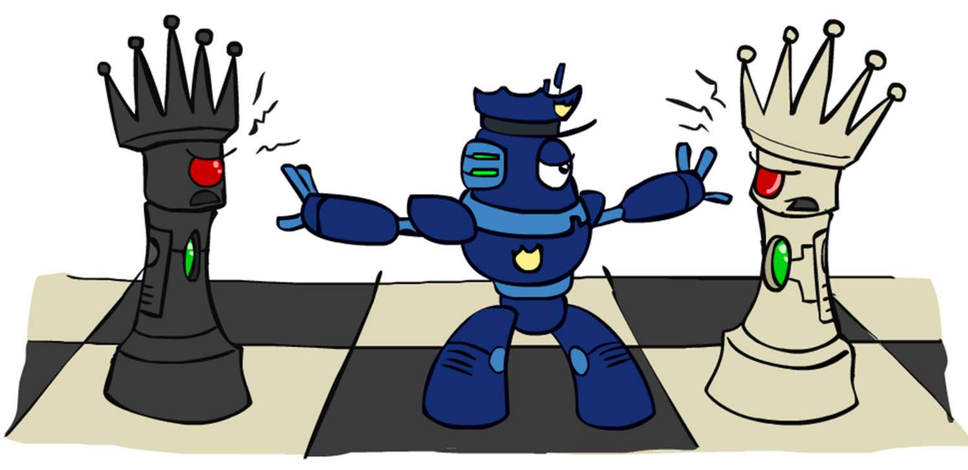
- **Variables:** Q_k
- **Domains:** {1,2,3,…,N}
- **Constraints:**
  - $\forall i,j$ non-threatening(Q_i,Q_j)



Types of Constraints

- **Constraints:**
  - Unary constraints involve a single variable (equivalent to reducing domains)
    - SA != green
  - Binary constraints involve pairs of variables
    - SA != WA
  - Higher-order constraints involve 3 or more variables
    - Crypt-arithmetic column constraints
- **Preferences (Soft Constraints):**
  - **Example:** red is better than green
  - Often representable by a cost for each variable assignment
  - These are called: **constrained** optimization problems

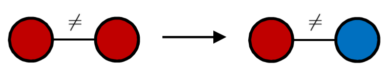
Iterative Improvement



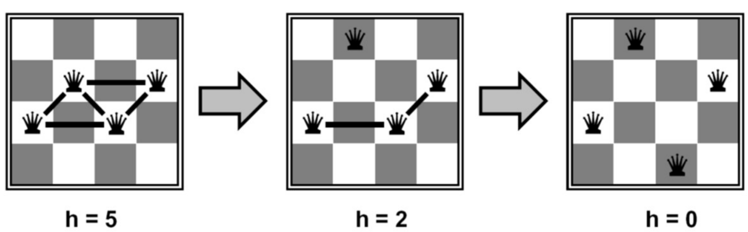
Iterative Improvement Algorithms for CSPs

- **Local search methods:** typically work with "complete" states, i.e., all variables assigned

- To apply to CSPs (Concept):
  - **Initial:** Start with fully assigned (mostly random) values to all variables
  - **While not solved:**
    - **Selection:** Take an assignment with unsatisfied constraints
    - **Improve:** reassign variable values that violates the fewest constraints (Iterative Min. Conflict, Hill Climbing)



Iterative Min Conflict - Example: 4-Queens



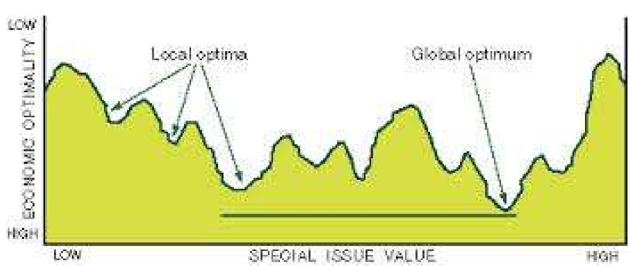
- **States:** 4 queens in 4 columns (4^4 = 256 states)
- **Operators:** move queen in column
- **Goal test:** no attacks
- **Evaluation:** c(n) = number of attacks

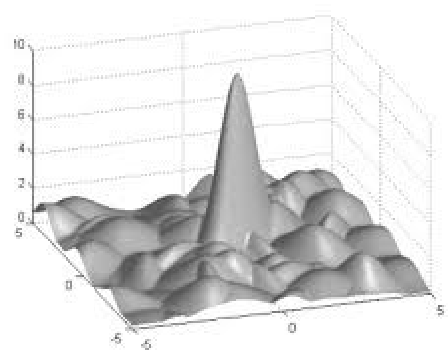
Hill Climbing

- Simple, general idea:
  - **Initial:** Start wherever
  - **Repeat:** move to the best neighboring state
  - **Stop:** If no neighbors better than current, quit
- What's bad about this approach?
  - Complete?
  - Optimal?
- What's good about it?

A typical genetic algorithm implementation

- In a complex problem space, local search will stop in local optimum
- If we have a population of candidate solutions, and perform local search on all of them, we have greater chance of finding the global best
- Population size <u>depends on</u> the <u>complexity</u> of problem space
  - In a convex space, even a single candidate and local search will be successful…

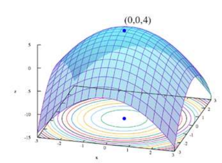




Optimization

Optimization (mathematics, computer science and operations research)

- **Mathematical Optimization/Programming:** techniques used for the selection of a best element (with some criterion) from the set of available alternatives
  - The best element gives the maximum or minimum of an objective function



- **Operations research:** a branch of mathematics that uses scientific techniques for decision making problems, attempts to find the best/optimal solutions or decisions.

- **Optimization algorithms:** are procedures executed iteratively by comparing various solutions till an optimum or a satisfactory solution is found.
  - Optimization algorithms help us to minimize or maximize an objective function E(x)

Optimization Algorithms

- Alpha-beta pruning
- Constraint satisfaction
  - AC-3 algorithm
  - Min conflicts algorithm
- Local search
  - Random-restart hill climbing
  - Tabu search
- Evolutionary computation
  - Evolution strategy
  - Genetic algorithms
  - Memetic algorithm
  - Swarm intelligence
    - Ant colony optimization
    - Bees algorithm
    - Particle swarm
- Harmony search (HS)
- Simulated annealing
- Gradient descent
- Cross-entropy method
- Combinatorial optimization:
  - Greedy randomized adaptive search procedure (GRASP)
  - Hungarian method
- Nonlinear optimization
  - BFGS method
  - Gauss–Newton algorithm
- Differential evolution
- Dynamic Programming
- Golden section search
- Branch and bound
- Interior point method
- Linear programming
  - Delayed column generation
  - Integer linear programming
  - Simplex algorithm
- Line search
- Minimax
- Nearest neighbor search (NNS)
- Newton's method in optimization

Metaheuristic Algorithms

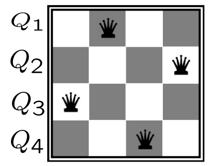
- Some of the optimization methods are of a family of methods called Metaheuristic.
- **Metaheuristic:** stochastic algorithms that use a combination of randomization and local search.
  - "A heuristic around heuristics"
  - An iterative approach, in which a heuristic is guided to explore and exploit a search space.
  - Usually designed for global optimization.
  - Non-Population Metaheuristics
    - Simulated Annealing, Tabu Search, VNS, GRASP, Iterated Local Search
  - Population Metaheuristics
    - Genetic Algorithims (GA), Particle Swarm Optimization (PSO), Ant Colony (AC)

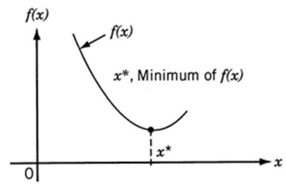
Pobulation Metaheuristics	Non-Pobulation Metaheuristics
Population of size M	Population of size 1
Recombining (crossover) and Perturbations (Mutation)	Only perturbations (small changes, similar to mutation)
Higher Complexity, but better performance	Less complexity and computation time

  - Tradeoff between complexity and performance

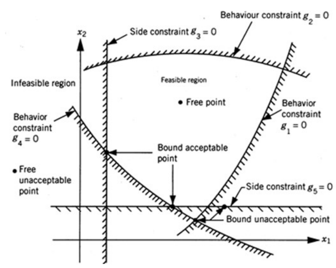
Discrete vs. Continuous – Single objective vs Multi Objective

- Sometimes we want to find the best values of a discrete variables (i.e. discrete domain) while Other times we search for the optimum value of a continuous variable (i.e. continuous domain)





- Sometimes we have more than one objective (i.e. we need to consider more than one objective when finding the best)



Stochastic Search Algorithms

- **Covered:**
  - Genetic Algorithms
  - Simulated Annealing
  - Tabu Search
  - Ant Colony Algorithm
  - PSO (Particle Swarm Optimization)
  - Harmony Search