# Constraint Satisfaction Problems: Local Search

Alice Gao Lecture 7

Based on work by K. Leyton-Brown, K. Larson, and P. van Beek



Learning Goals

Introduction to Local Search

Local Search Algorithms Hill climbing

Revisiting the Learning goals

By the end of the lecture, you should be able to

- Describe the advantages of local search over other search algorithms.
- Formulate a real world problem as a local search problem using complete-state formulations.
- Given a local search problem, verify whether a state is a local optimum.

# Why Use Local Search?

- Many search spaces are too big for systematic search.
- For CSPs, we only need to find a goal node. The path to a goal is irrelevant.
- Solution: local search

### What is local search?

- Keep track of a single node, which is a complete assignment of values to variables.
- Move to a neighbour of the node based on how good the neighbour is.

### When should we use local search?

- The state space is large or infinite.
- Memory is limited.
- To solve pure optimization problems with a fitness function but no goal test.

# Local Search

For solving a CSP, a local search problem consists of a:

- A set of variables
- Domains for the variables
- Constraints on the joint values of the variables
- A node in the search space is a complete assignment to *all* of the variables.
- A neighbour relation: which nodes do I explore next?
- A cost function: how good is each assignment? In which direction should we go?

Example: 4-Queens Problem

# Example: Traveling Salesperson Problem

#### Learning Goals

Introduction to Local Search

#### Local Search Algorithms Hill climbing

Revisiting the Learning goals

# Questions

The problem formulation:

- What is the neighbour relation?
- What is the cost function?

Executing the algorithm:

- Where do we start?
- Which neighbour do we move to?
- When do we stop?

Properties and performance of the algorithm:

- Given enough time, will the algorithm find the global optimum solution?
- How much memory does it require?
- How does the algorithm perform in practice?

# Hill climbing

#### Where do we start?

Start with a random or good solution.

Which neighbour do we move to?

Move to a neighbour with the lowest cost. Break ties randomly. Greedy: does not look ahead beyond one step.

When do we stop?

Stop when no neighbour has a lower cost.

How much memory does it require?
Only need to remember the current node.
No memory of where we've been.

### Hill climbing in one sentence

#### Climbing Mount Everest in a thick fog with amnesia

**CQ:** Will hill climbing find the global optimal solution given enough time?

- (A) Yes. Given enough time, hill climbing will find the global optimal solution for every problem.
- (B) No. There are problems where hill climbing will NOT find the global optimal solution.

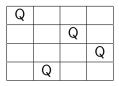
# Dealing with plateaux

- Allow sideway moves to escape a shoulder.
- An infinite loop when we are on a flat local optimum.
- Limit the number of consecutive sideway moves.

Tabu search: keep a small list of recently visited states and forbid the algorithm to return to those states.

# CQ: Is this state a local optimum?

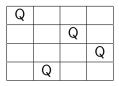
**CQ:** Consider the following state of the 4-queens problem. Consider successor function B: swap the row positions of two queens. Is this state a local optimum?



(A) Yes(B) No(C) I don't know.

# CQ: Is this state a local optimum?

**CQ:** Consider the following state of the 4-queens problem. Suppose that we use successor function A: move a single queen to another square in the same column. Is this state a local optimum?



(A) Yes(B) No(C) I don't know.

# Choosing the Neighbour Relation

How do we choose the neighbour relation?

Small incremental change to the variable assignment

There's a trade-off:

- bigger neighbourhoods:
- smaller neighbourhoods:

# Revisiting the Learning Goals

By the end of the lecture, you should be able to

- Describe the advantages of local search over other search algorithms.
- Formulate a real world problem as a local search problem using complete-state formulations.
- Given a local search problem, verify whether a state is a local optimum.