Lecture

Foundations of Artificial Intelligence

Part 6 – Local Search

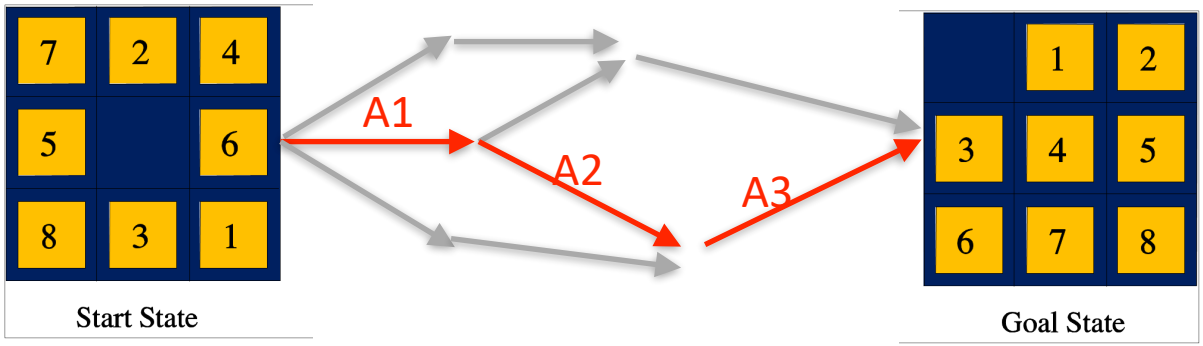
Dr. Mohsen Mesgar

Universität Duisburg-Essen

Recall …

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken



- **Uninformed search**
  - BFS: Breadth-first search
  - DFS: Depth-first search
  - ID: Iterative deepening

◊ **Informed search** using heuristics

◊ Greedy Best-first search

◊ A* search

Any other open questions?

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Where are we?

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Intelligence & Agents
- Search
  - Uninformed
  - Informed
  - **Local**
- Uncertainty & Learning
- Learning to Represent

Local Search

Optimization Problems

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- AI is about setting a goal and optimize the actions to achieve the goal

Optimization Problems

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- AI is about setting a goal and optimize the actions to achieve the goal
- There are two major optimization problems
  - **Optimal path finding:** start and goal states are clearly defined

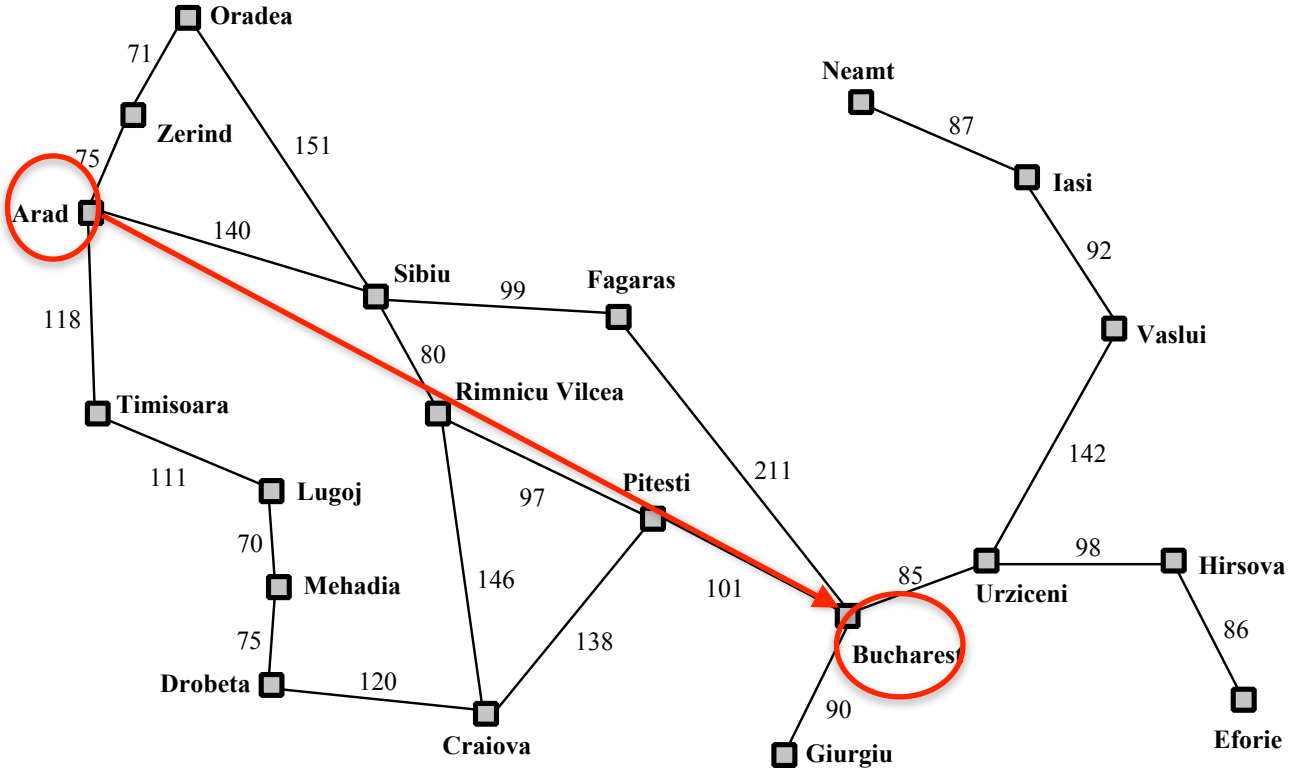
Optimization Problems

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- AI is about setting a goal and optimize the actions to achieve the goal
- There are two major optimization problems
  - **Optimal path finding:** start and goal states are clearly defined
  - **Optimal goal state finding:** start state is given but the goal state is unknown. Instead we **have a set of constraints that should be satisfied**

Path Finding

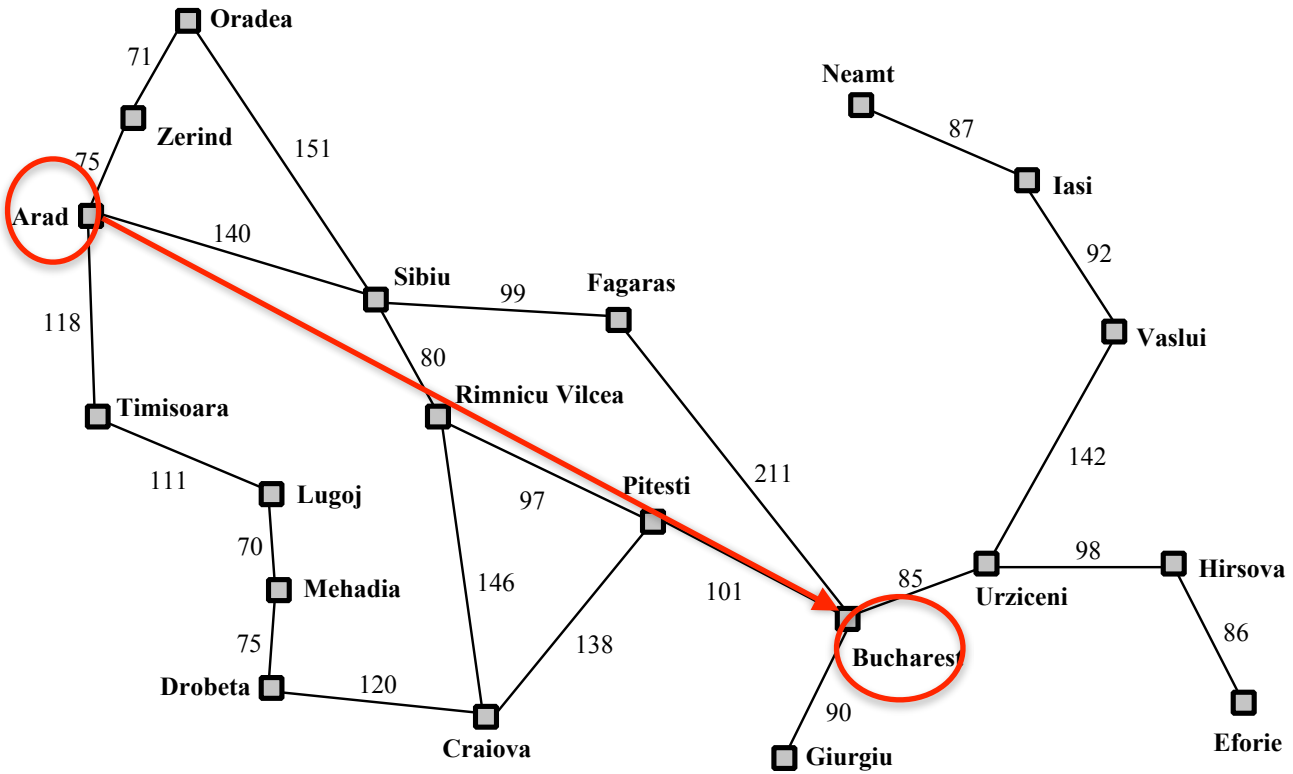


Path Finding

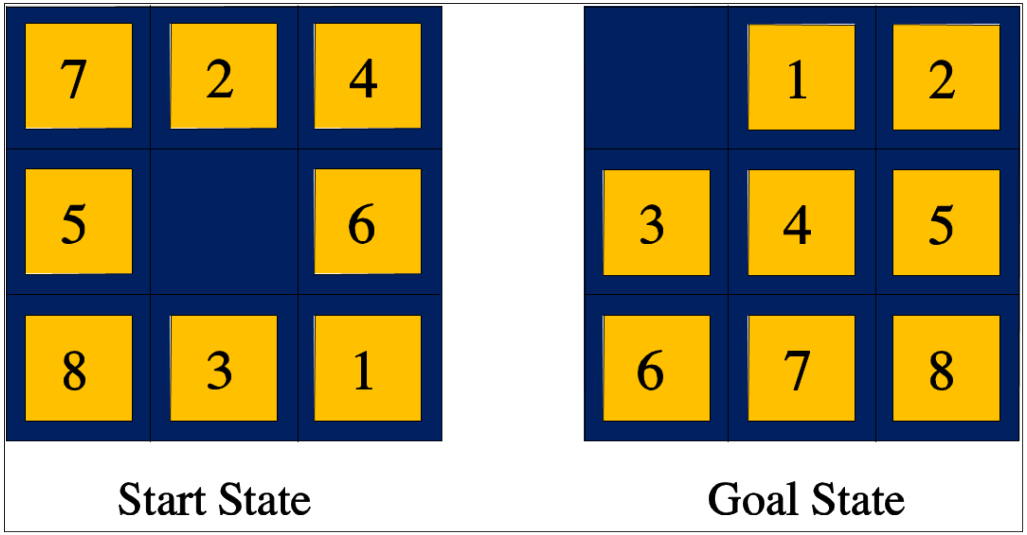
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Optimal path finding



8-Puzzle

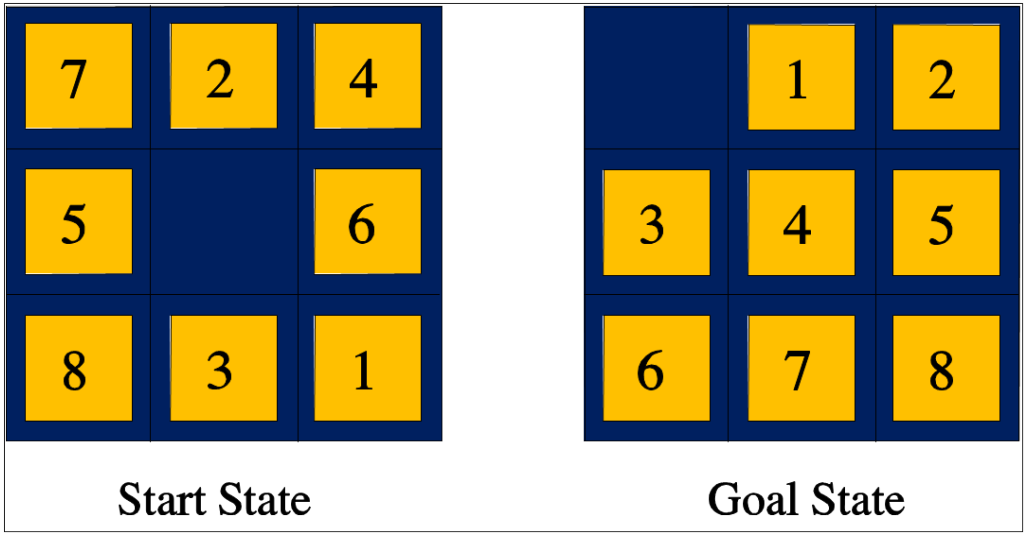


8-Puzzle

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Optimal path finding



Sudoku

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- **Goal:** To fill a 9 × 9 grid with digits so that each column, each row, and each of the nine 3 × 3 subgrids that compose the grid contain all of the digits from 1 to 9

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Sudoku

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- **Goal:** To fill a 9 × 9 grid with digits so that each column, each row, and each of the nine 3 × 3 subgrids that compose the grid contain all of the digits from 1 to 9

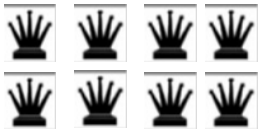
5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Optimal goal state finding

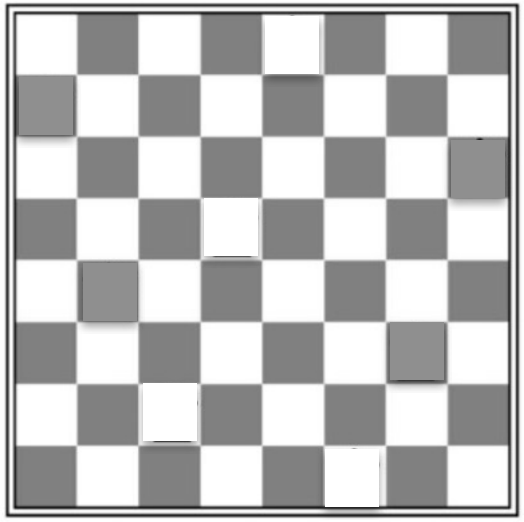
The N-Queens Problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken



- **Goal:** To put eight queens on a N*N chess-board such that none of them threatens any of others.
  - No two queens share the same row,
  - No two queens share the same column
  - No two queens share the same diagonal.

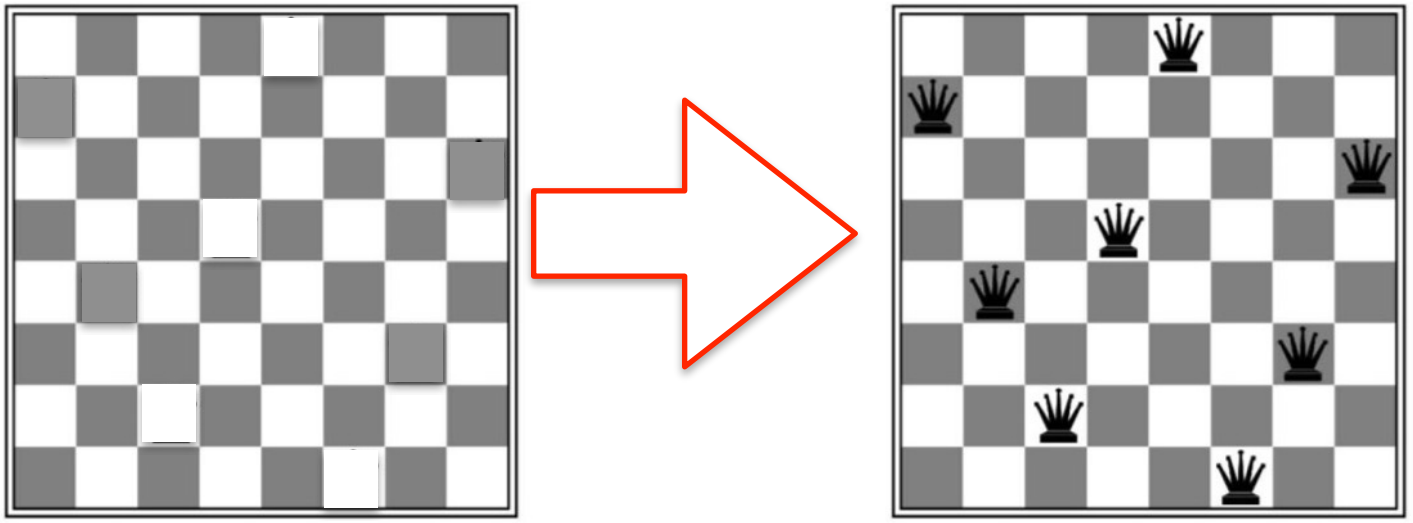


The 8-Queens Problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- There are 92 goal states. We should find one of them.

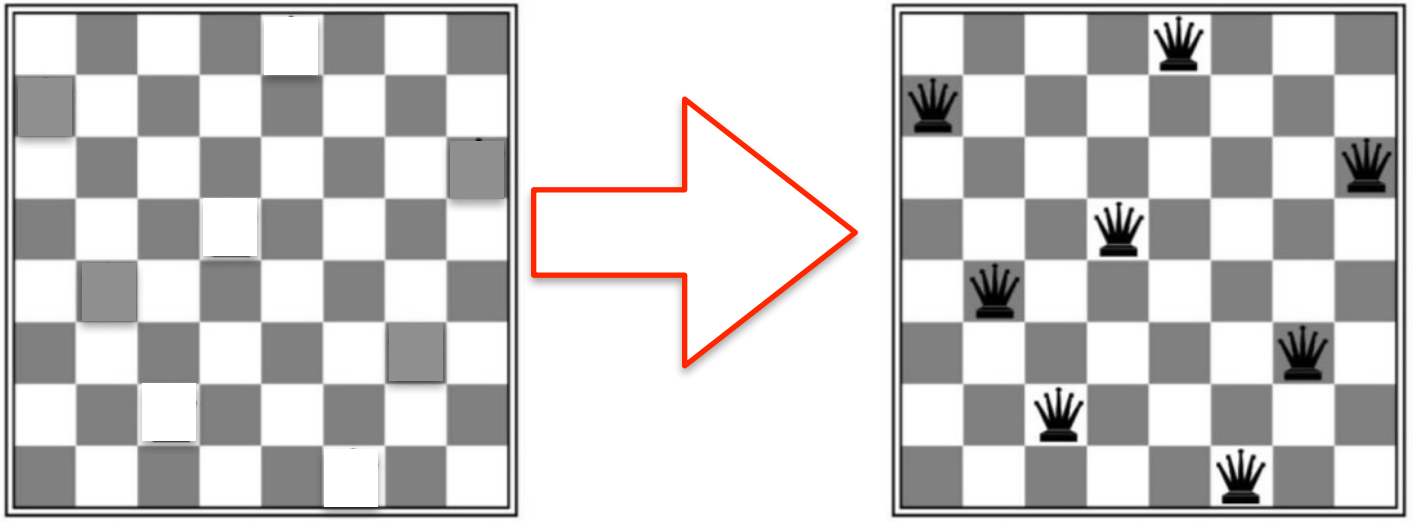


Example: The 8-Queens Problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Optimal goal state finding



Optimization Problems

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- There are two major optimization problems
  - **Optimal path finding:** start and goal states are clearly defined
    - Agent uses uninformed or informed search algorithms
  - **Optimal goal state finding:** start state is given but the goal state is unknown. Instead we **have a set of constraints that should be satisfied**
    - Agent uses local search algorithms

Local search algorithms

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

State space:

- A set of **"complete" configurations**

Goal :

- Is a **configuration** that **satisfies all constraints**

Local search

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Approach

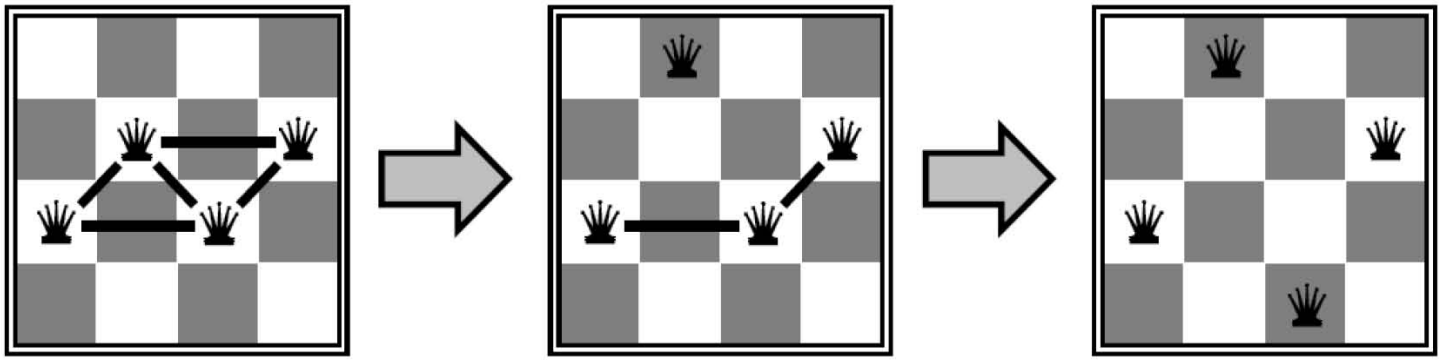
- **keep a single "current" state** (or a fixed number of them)
- try to improve it by **maximizing a heuristic function**
- use only **“local” improvements**
  - i.e., only **modifies the current state(s)**

Example: n-queens problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Put 4 queens randomly on the chessboard
- move one queen so that it reduces the number of conflicts
- often solves n-queens problems almost instantaneously for very large problems
  - e.g. n = 1,000,000 can be easily solved



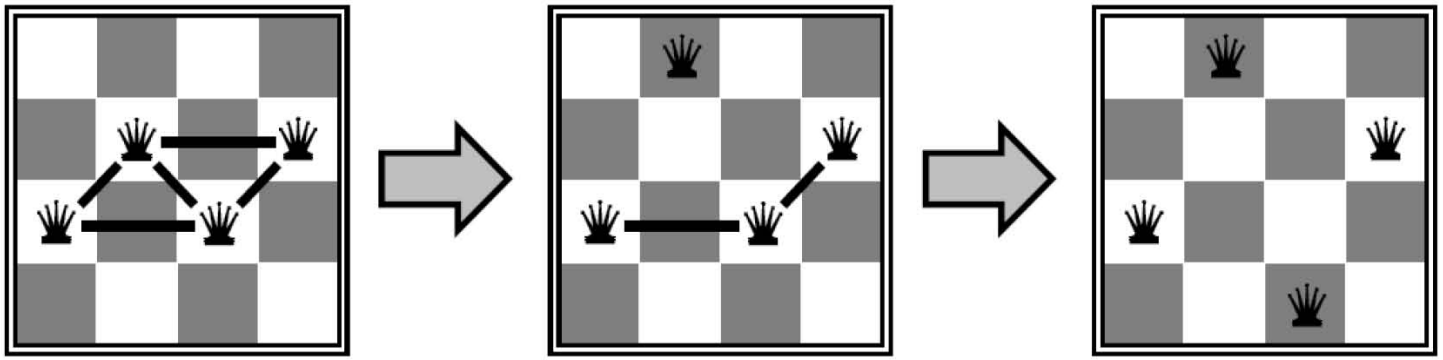
h = 5 h = 2 h = 0

Example: n-queens problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Put 4 queens randomly on the chessboard
- move one queen so that it reduces the number of conflicts
- often solves n-queens problems almost instantaneously for very large problems
  - e.g. n = 1,000,000 can be easily solved



h = 5 h = 2 h = 0

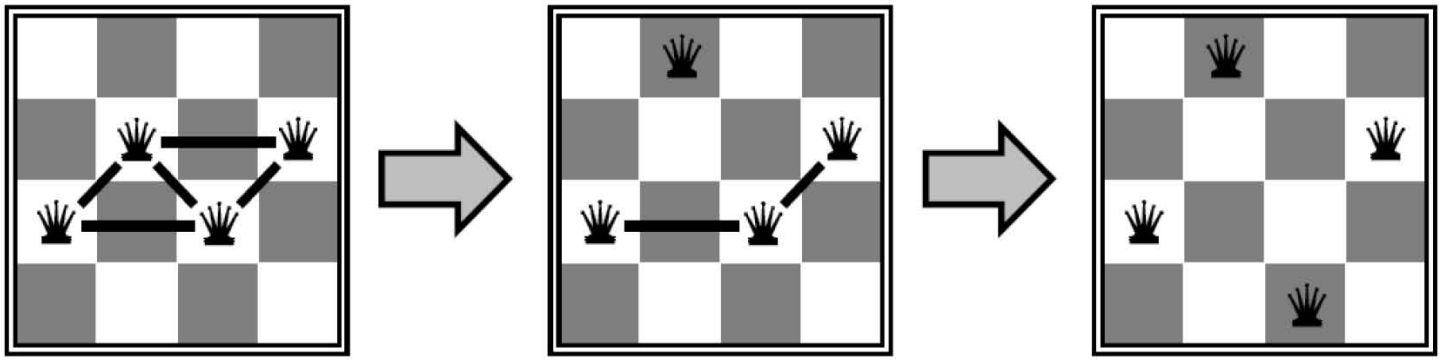
It's like we are minimizing a function that gives the number of conflicts

Example: n-queens problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Put 4 queens randomly on the chessboard
- move one queen so that it reduces the number of conflicts
- often solves n-queens problems almost instantaneously for very large problems
  - e.g. n = 1,000,000 can be easily solved



h = 5 h = 2 h = 0

It's like we are minimizing a function that gives the number of conflicts
Or we are maximizing a function that estimates the value of the state

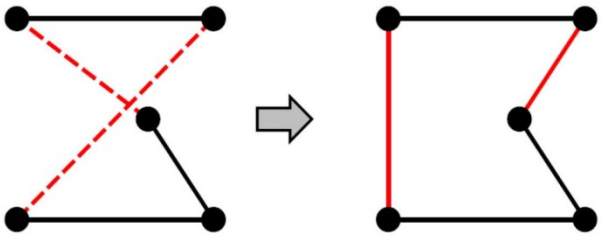
Example: Traveling salesperson problem

Example: Traveling salesperson problem

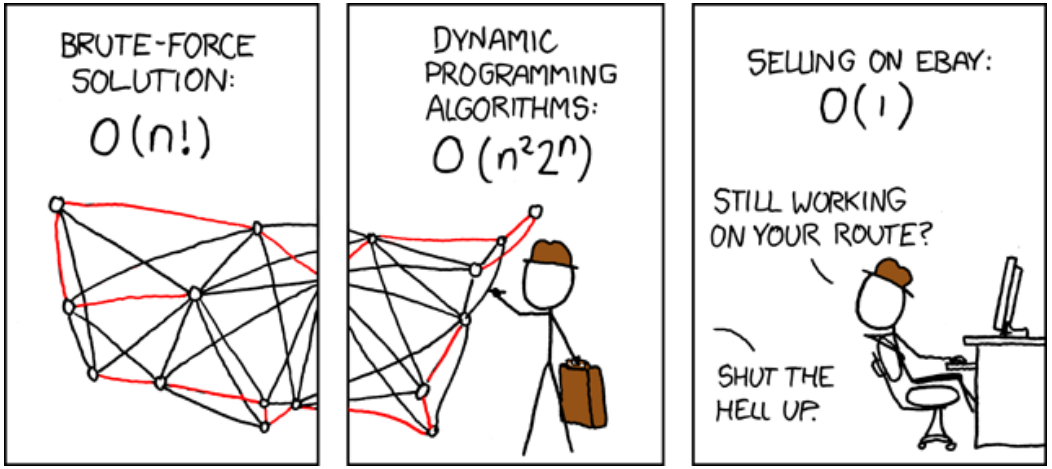
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- start with a complete tour that passes all cities
- perform pairwise exchanges so that the exchange reduces the cost of the tour
- variants of this approach quickly get close to an optimal solution even for thousands of cities



This will be our first challenge!

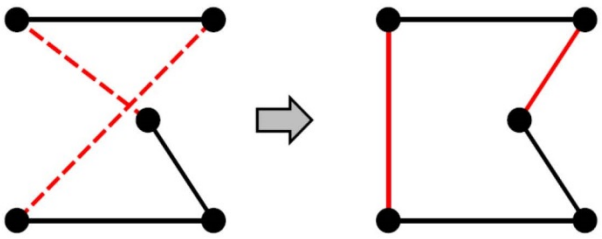


Example: Traveling salesperson problem

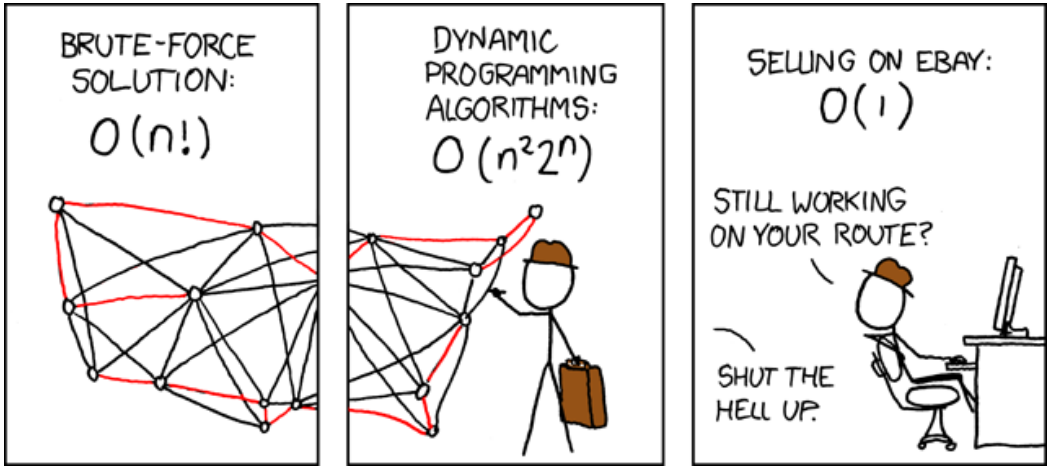
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- start with a complete tour that passes all cities
- perform pairwise exchanges so that the exchange reduces the cost of the tour
- variants of this approach quickly get close to an optimal solution even for thousands of cities



This will be our first challenge!



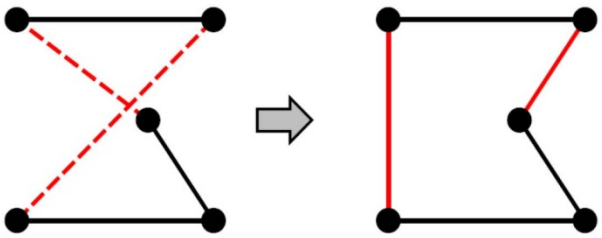
It's like we are minimizing the a function that gives cost of a tour

Example: Traveling salesperson problem

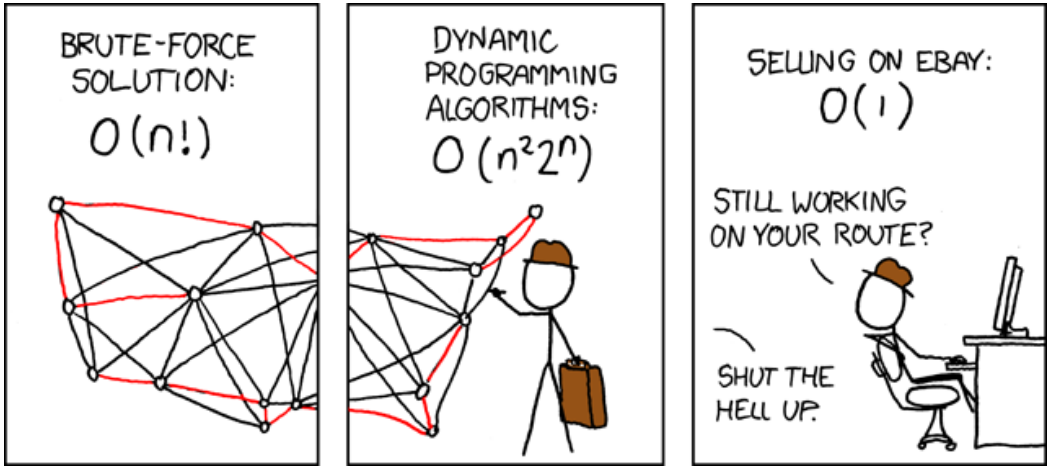
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- start with a complete tour that passes all cities
- perform pairwise exchanges so that the exchange reduces the cost of the tour
- variants of this approach quickly get close to an optimal solution even for thousands of cities



This will be our first challenge!



It's like we are minimizing a function that gives cost of a tour

Or like we are maximizing a function that estimates the value of the state

Local search

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Advantages

- It uses very **little memory —> we keep only one or a few states**
- often **quickly finds solutions in large or infinite state spaces**

Disadvantages

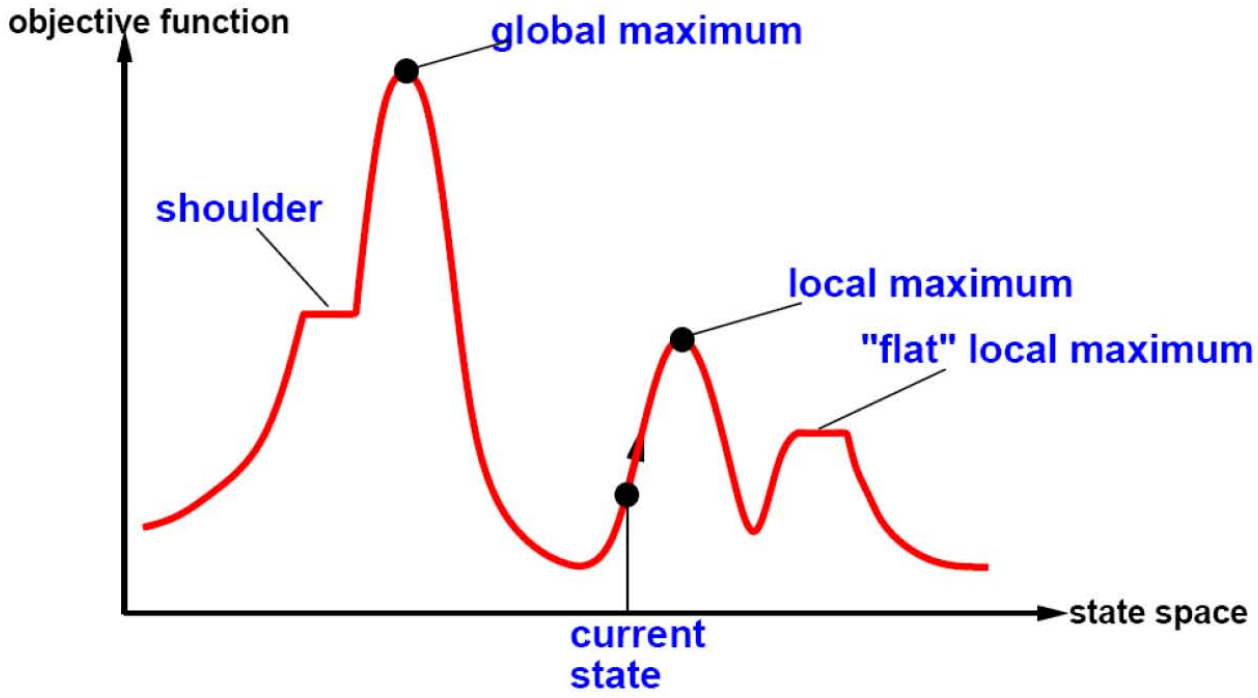
- **no guarantees for completeness or optimality**

State space landscape

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

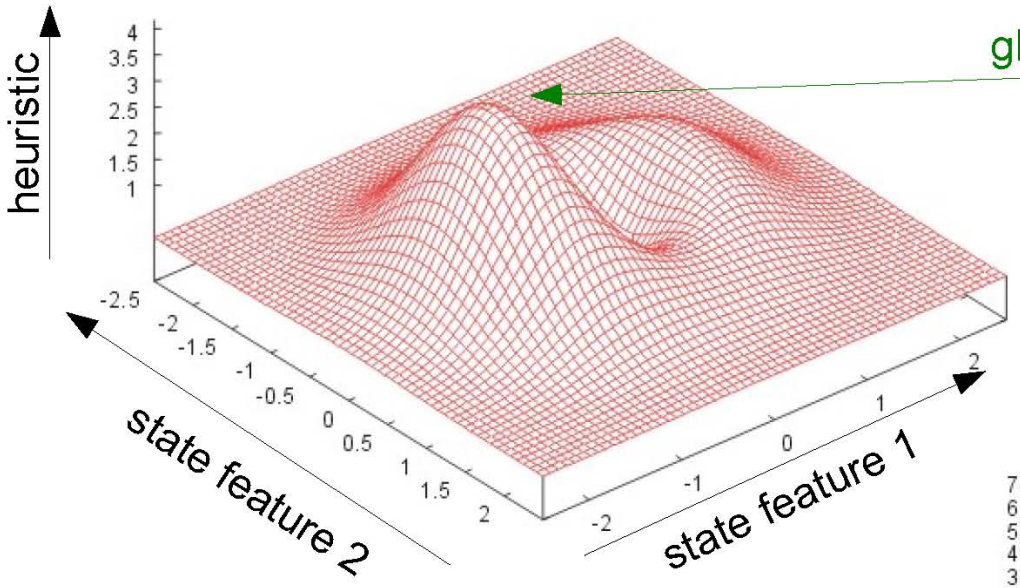
- location: states
- elevation: value of objective function
- problematic regions: local maximum, plateau

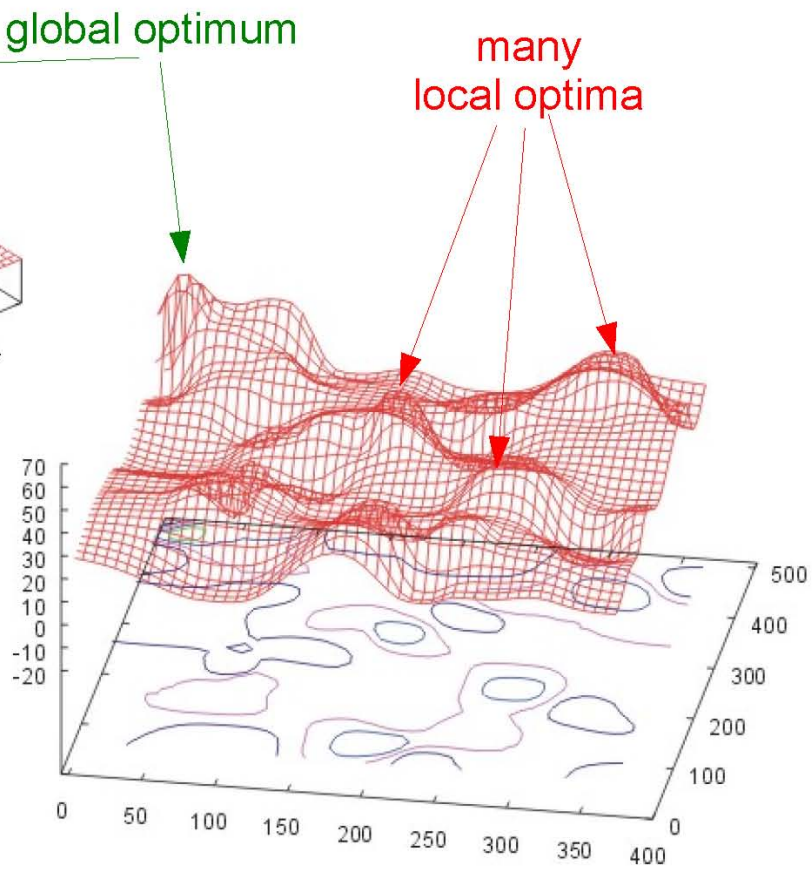


Multi-dimensional state landscape

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken





Local search algorithms

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Hill climbing
- Beam search
- Simulated annealing
- Genetic algorithms

Hill Climbing

Hill climbing search (greedy local search)

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Expand the current state → generate all neighbor states
- Neighbor states: states that can be reached within a single transformation of the current state
- Move to the one with the highest evaluation.
- Continue as long as the evaluation score increases.

https://c1.staticflickr.com/8/7140/7486564384_695593d27a_b.jpg

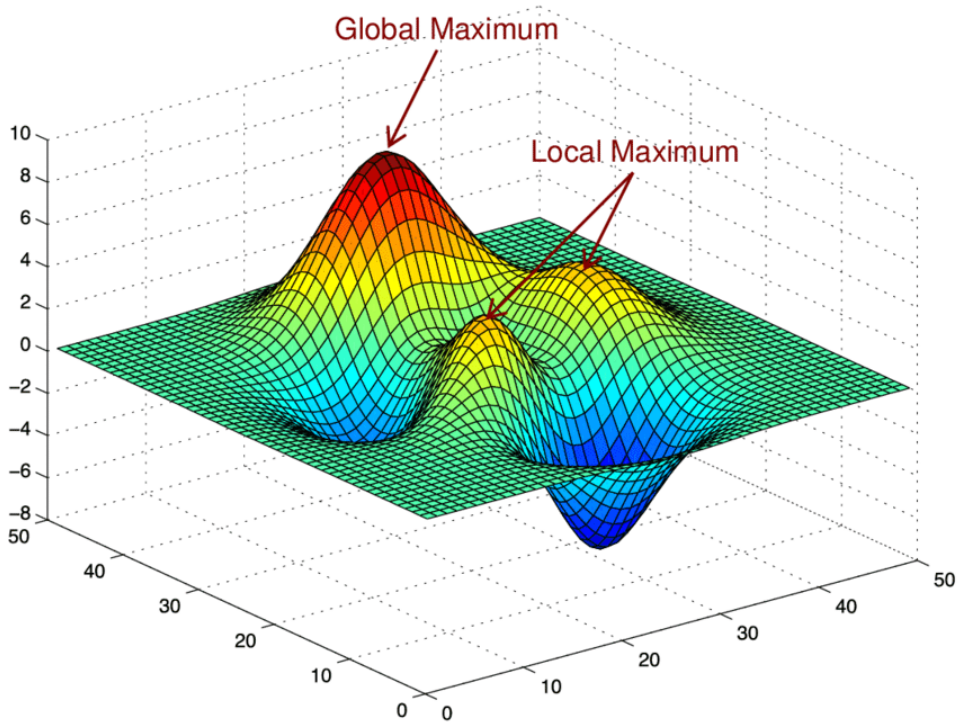
Hill climbing search

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Main problem: Local optimum

- The algorithm will stop as soon as it is at the top of a hill.
- But it is actually looking for a mountain peak.



https://www.researchgate.net/figure/11-Illustration-of-local-optimum-and-global-optimum_fig16_306558608

8-queens problem

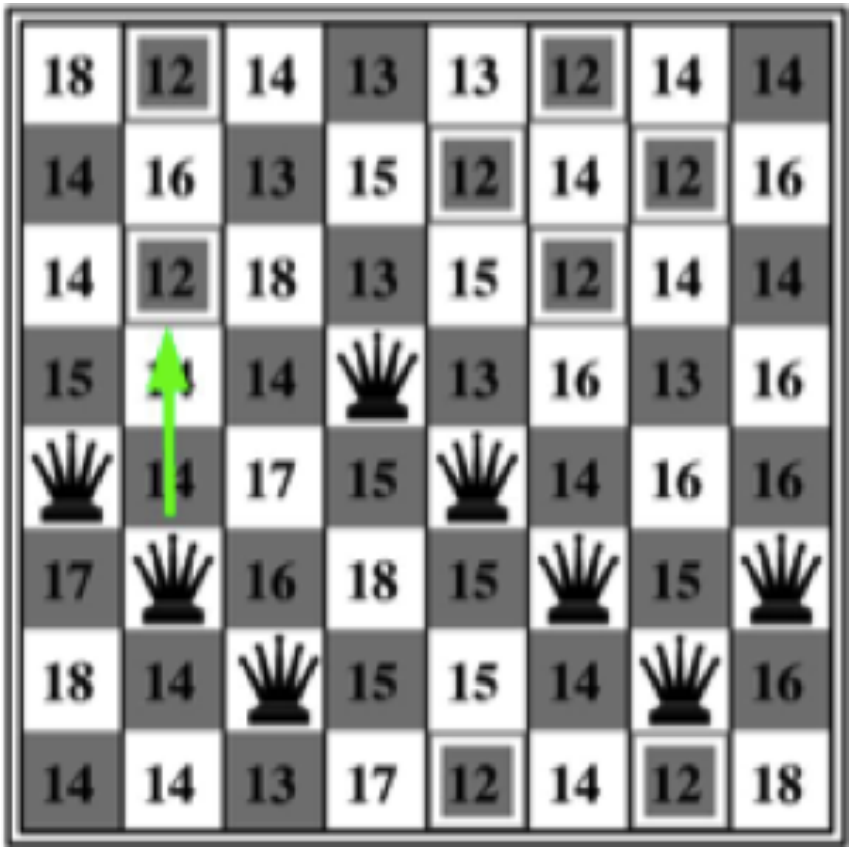
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Heuristic h as the evaluation function

- Number of pairs of queens that attack each other → minimize h
- Example state: h=17
- Best neighbor: h=12

  → Several possible moves



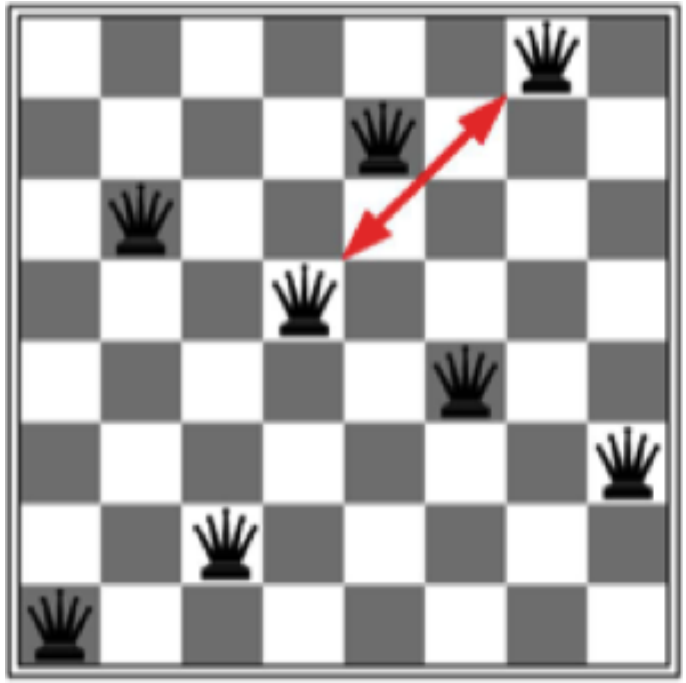
Example: 8-queens problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Stuck in local optimum with $h=1$

- No queen can move without decreasing the number of attacked pairs



Randomized hill climbing variants

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Random restart hill climbing

- **Different initial states** result in **different local optimum**

→ We should run **several iterations** with **different starting states**

Stochastic hill climbing

- Select the **successor state randomly based on the** probability of each state
- The probability of a state is estimated using the heuristic value for the state

Beam Search

Beam search

UNIVERSITÄT DUISBURG ESSEN

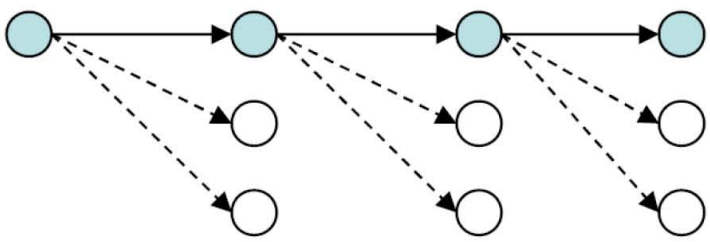
Offen im Denken

Idea:

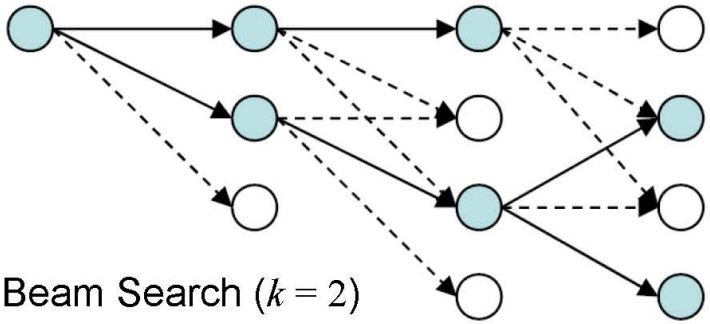
- **Keep track of k states** rather than **just one**
- k is called **the beam size**

Algorithm

- Start with **k randomly generated states**
- At each iteration, generate **all successors of all k states**
- If any state **satisfies the constraints**, **stop**;



Hill-Climbing Search



Beam Search ($k = 2$)

Beam search

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Implementation

- similar to the **tree search algorithms**
- sort the **queue** by **the heuristic function h** (as in greedy best-first search), but **limit the size of the queue to k**

→ expand all nodes in queue simultaneously (not in sorted order)

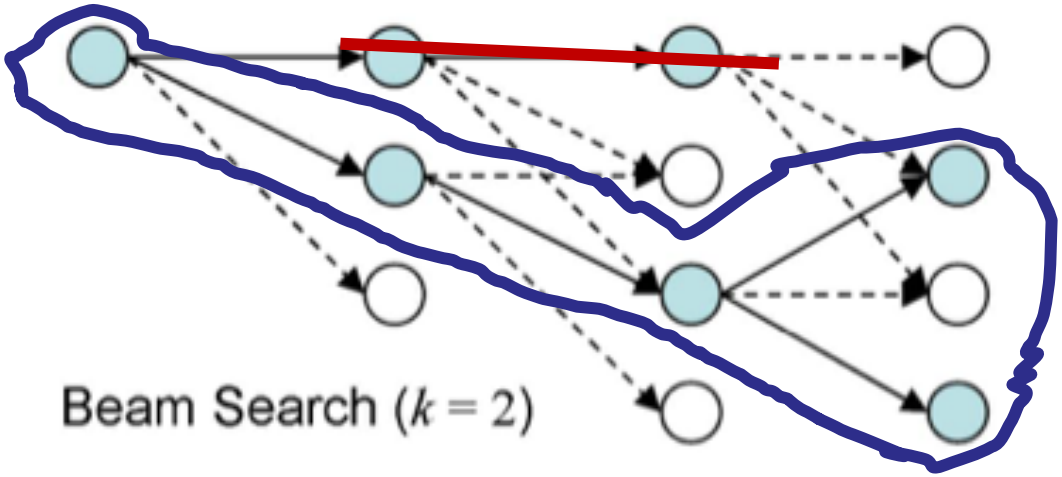
Beam search

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Note

- **beam search** is **different** from **k** **parallel hill-climbing** searches!
- information from different beams is combined



Beam search

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Effectiveness

- suffers from **lack of diversity of the k states**
  - e.g., if one state has better successors than all other states, alternative routes are pruned early
  - **often** not more effective than basic hill climbing

Stochastic beam search

- chooses k **successors at random**
- selection probability of each states depends on heuristic value for the state

Simulated Annealing

Simulated annealing

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Algorithm:

- temporarily accept worse configurations but with decreasing probability

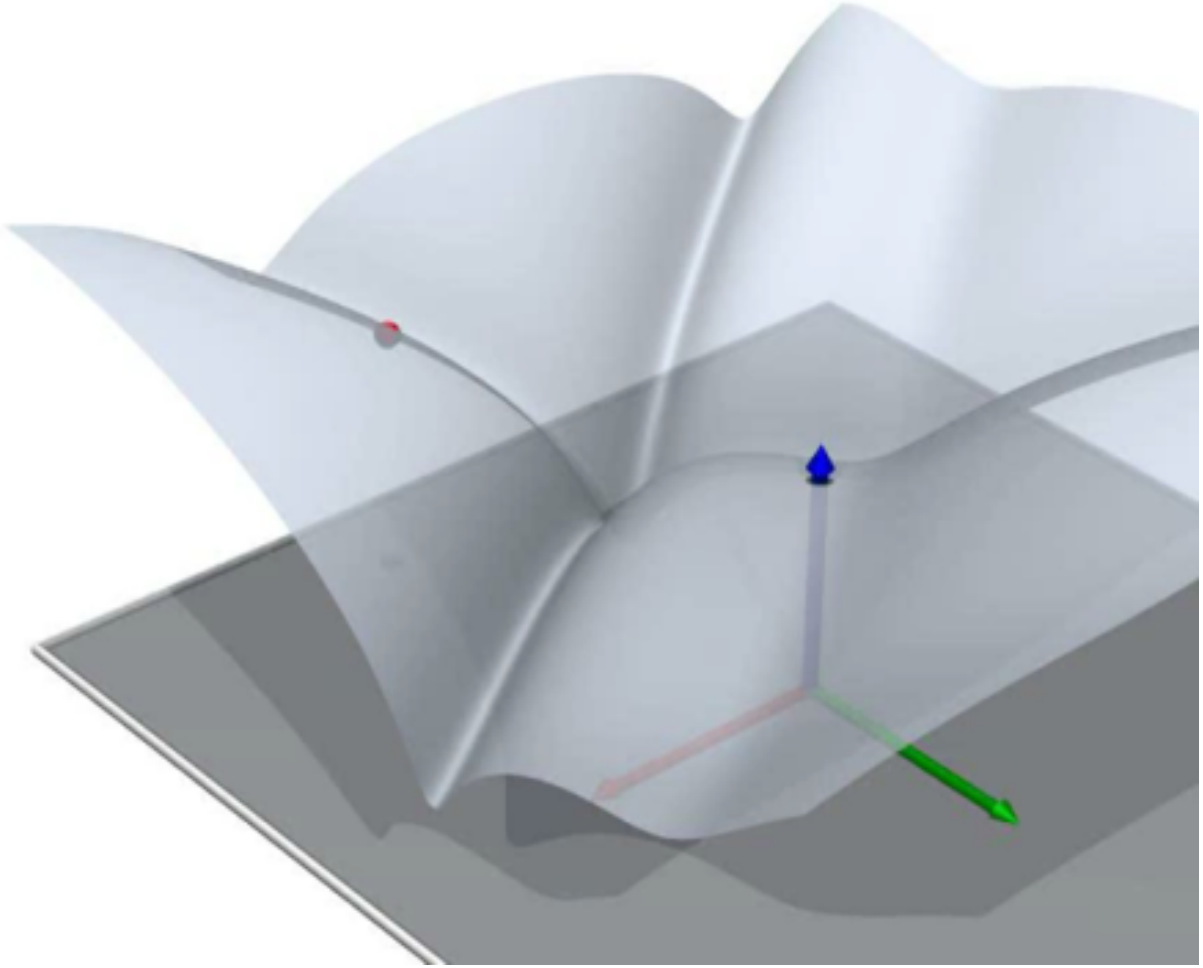
- Combination of **hill climbing** and **random walk** through the state space
  - **escape local maximum** by **allowing some "bad" moves**
  - but gradually **decrease their frequency** (the *temperature*)

Effectiveness

- it can be proven that if **the temperature is lowered slowly enough**, the **probability of converging to a global optimum approaches 1**.

Widely used in VLSI layout, airline scheduling, etc.

Simulated annealing



Genetic Algorithms

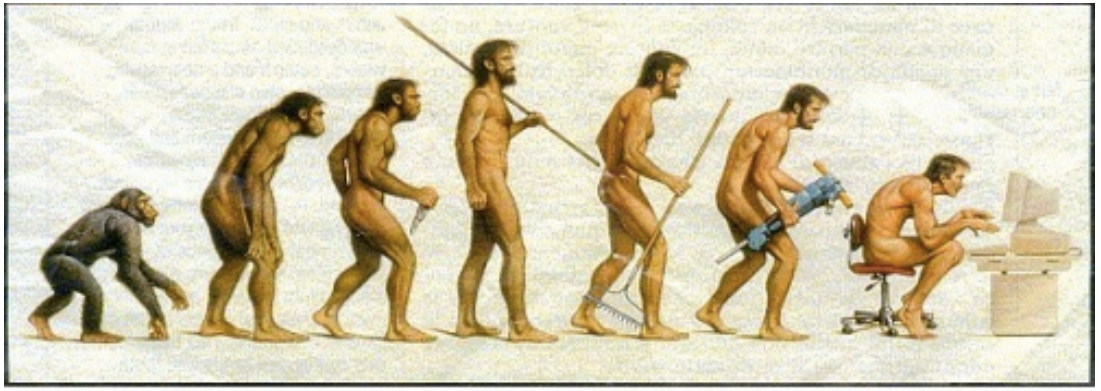
Metaphor: Genetic algorithms

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Analogy with evolutionary theory → Simulation of natural selection

- View **states** as **individuals in a population**
- **Objective function** measures the **fitness of an individual**
- Allow "**fittest**" individuals **to reproduce**
- **Eventually**, the **population** should evolve towards **fitter individuals.**

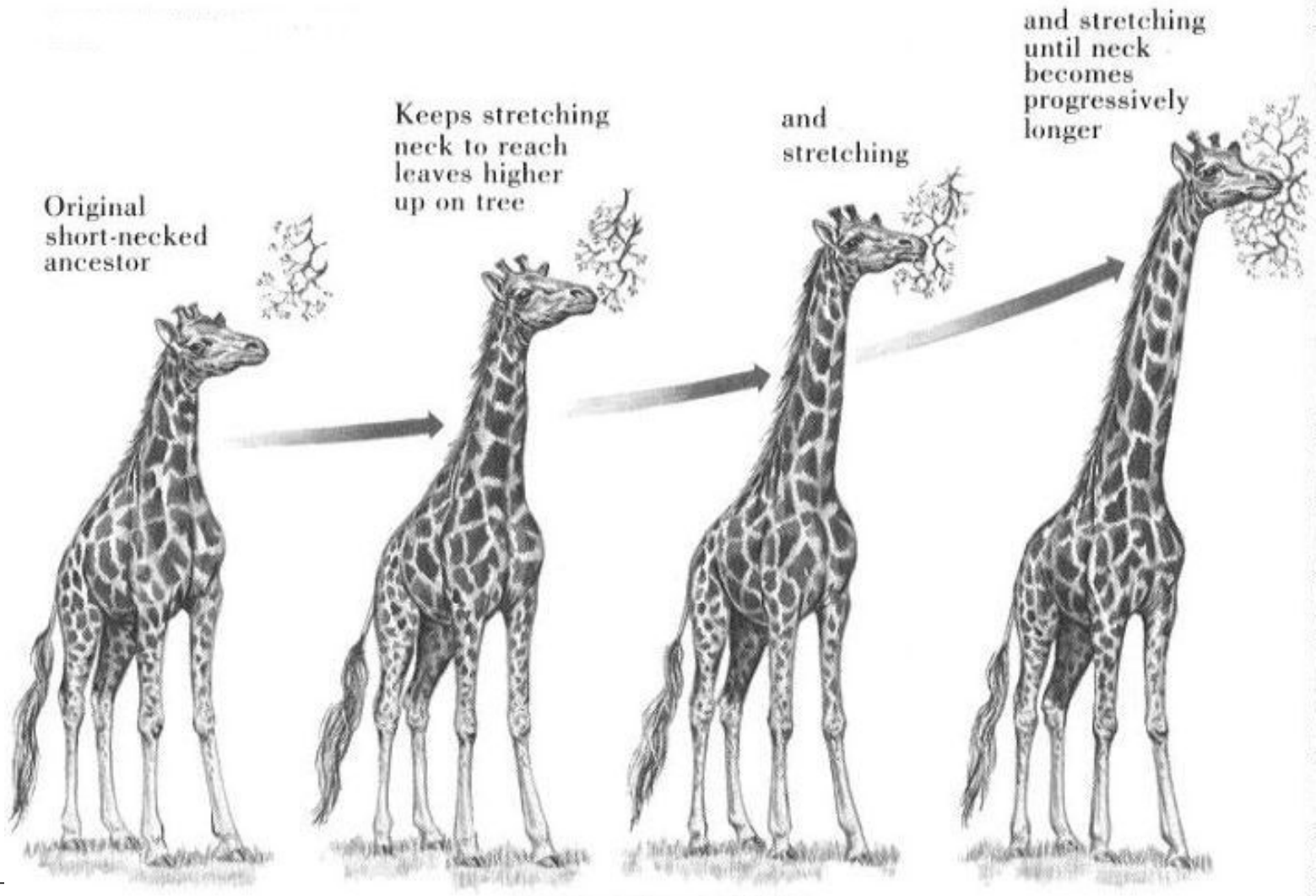


Somewhere, something went terribly wrong

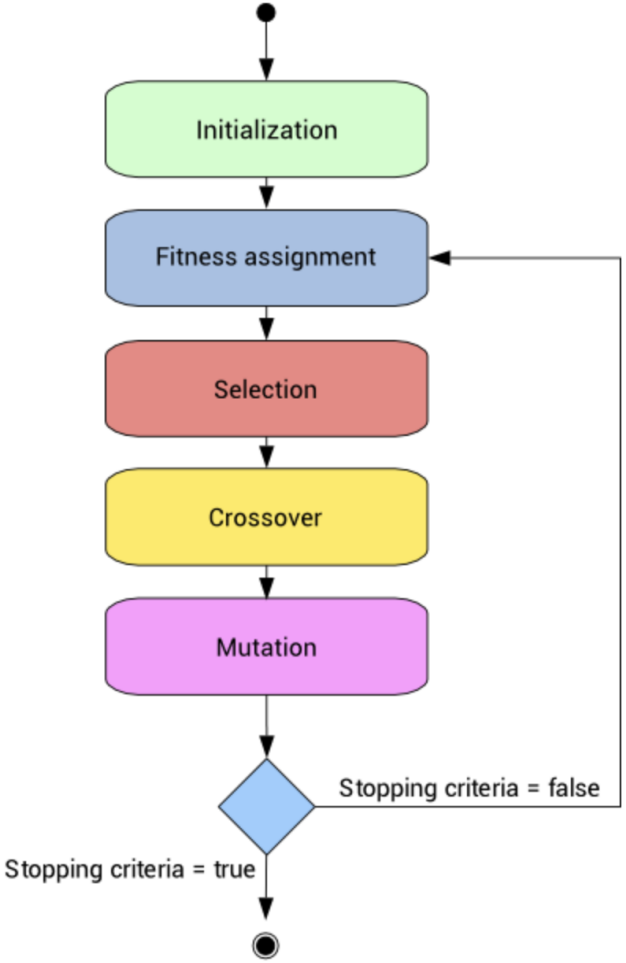
Genetic evolution

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken



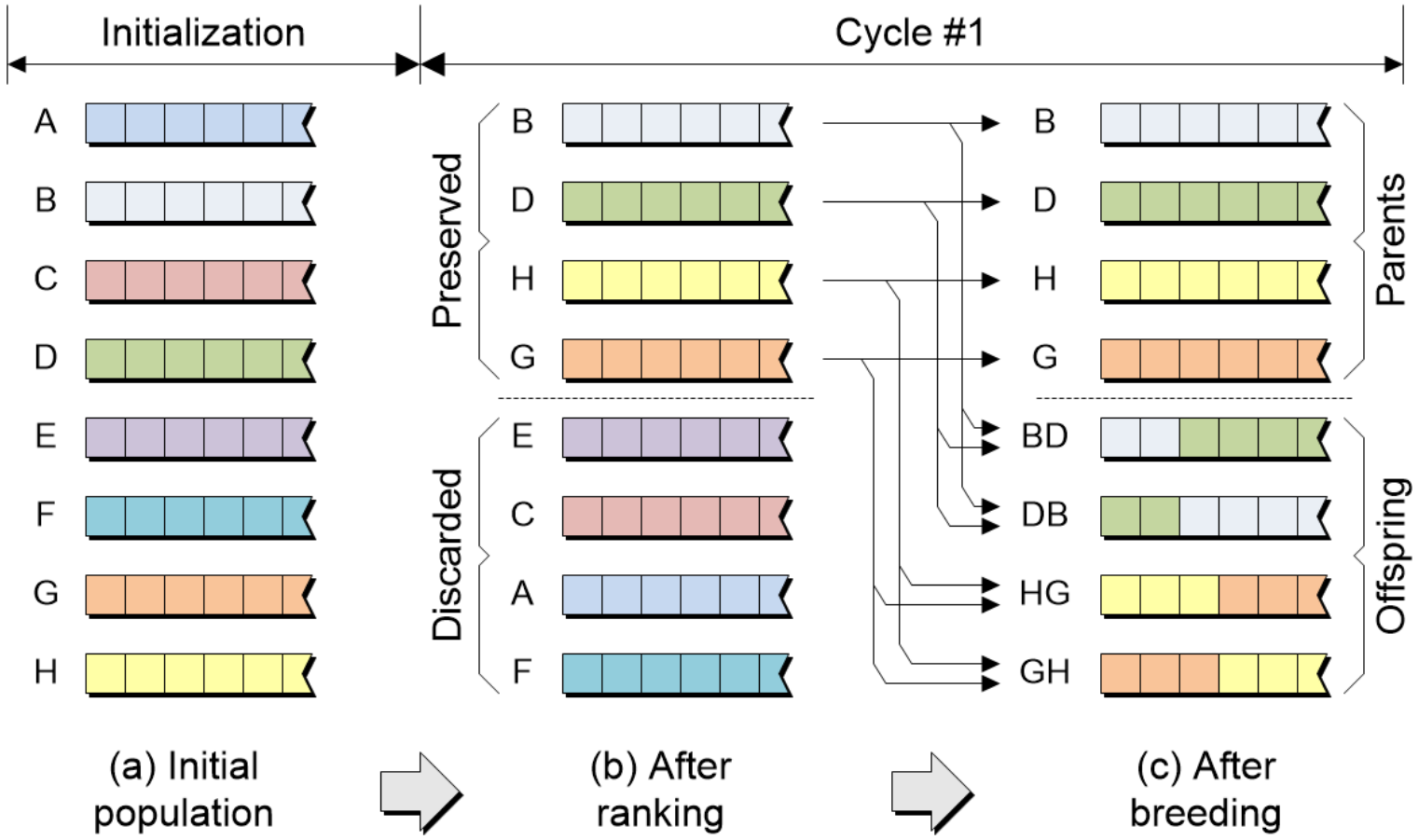
Genetic algorithm



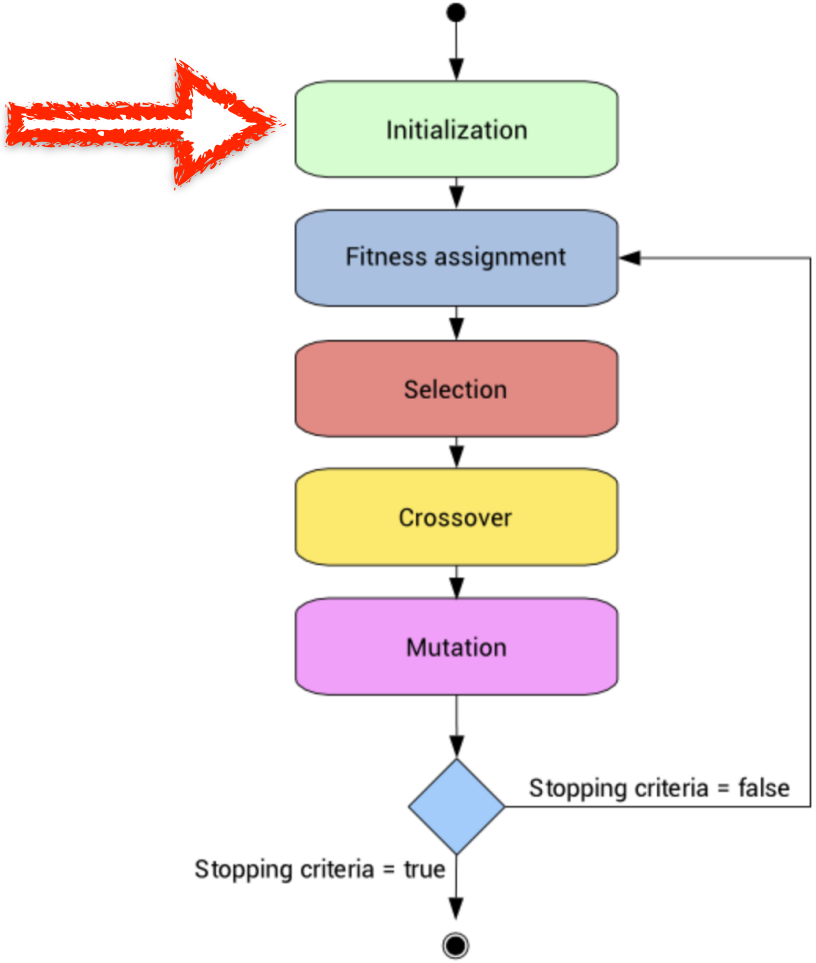
Genetic algorithm

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken



Genetic algorithm



State = problem configuration

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

The problem needs to be represented by a configuration.

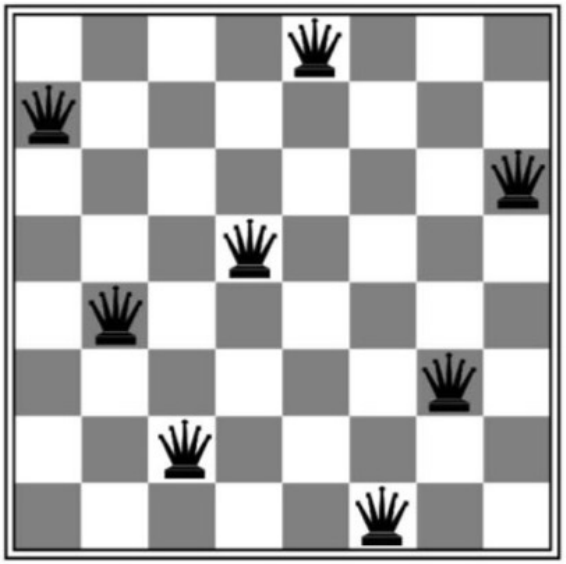
$X =$ | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

8 queen representation

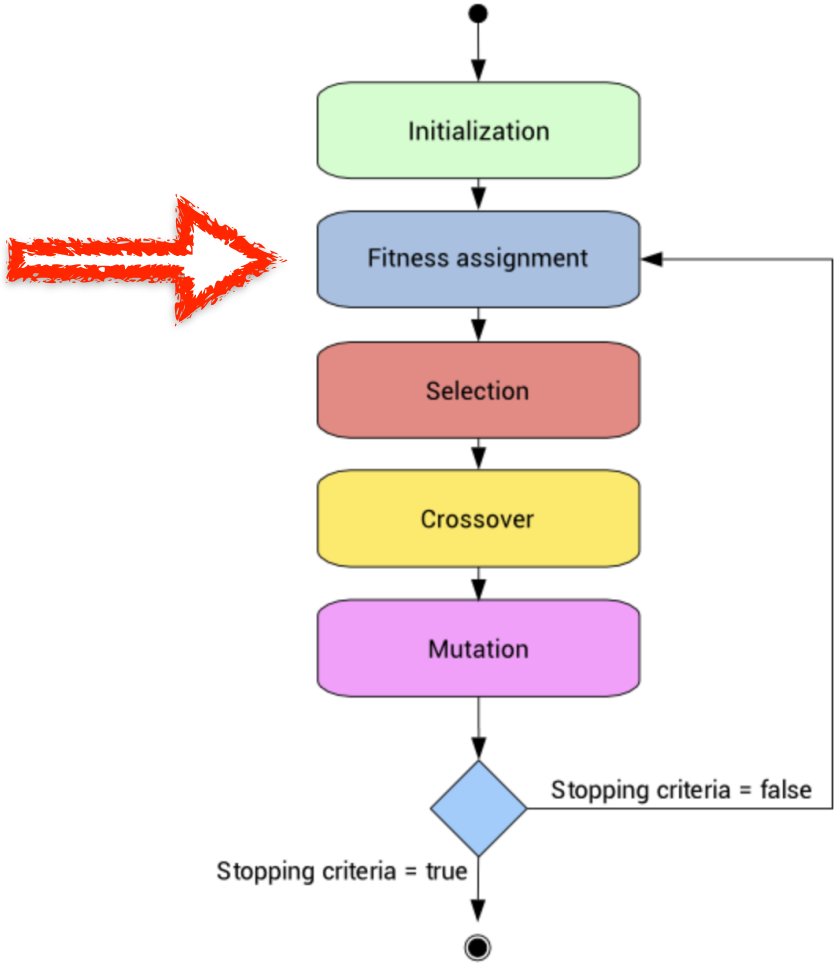
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

$$x = 74258136$$



Genetic algorithm



Fitness assignment and selection

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Start with **a random population as the initial state**
- Apply an **evaluation function (fitness)**
  - should assign **higher values to better states**
  - → Number of non-attacking pairs of queens
- **determine selection probability** according to fitness

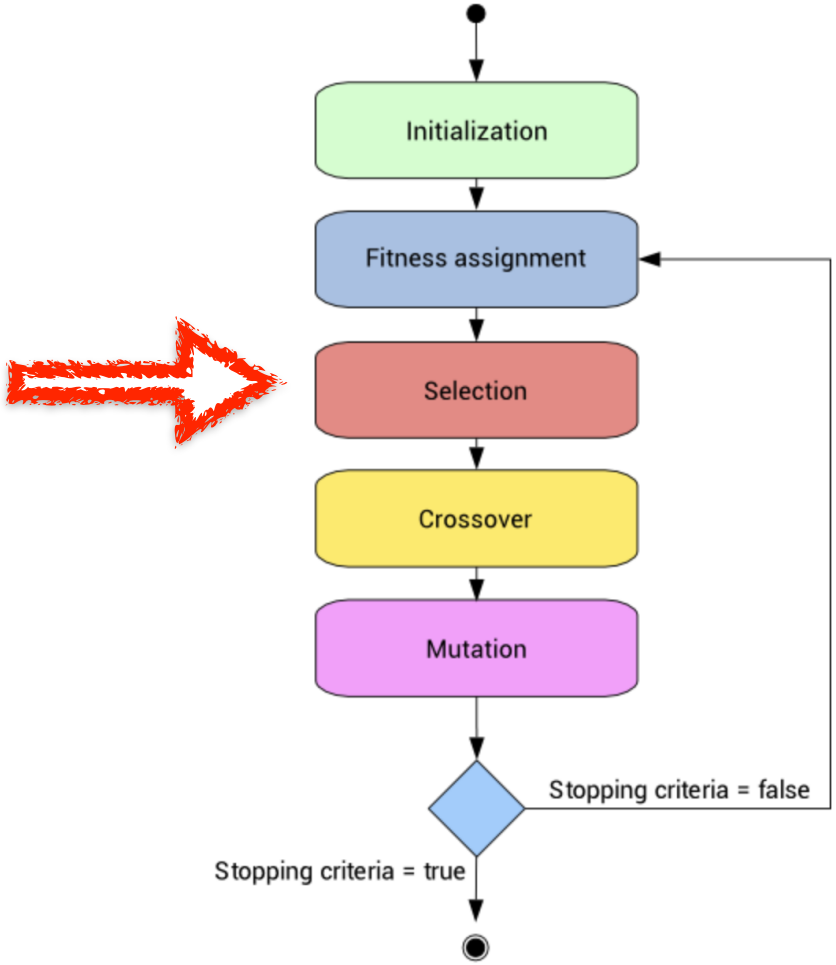
24748552 → 24 → 31%

32752411 → 23 → 29%

24415124 → 20 → 26%

32543213 → 11 → 14%

Genetic algorithm



Selection

UNIVERSITÄT DUISBURG ESSEN

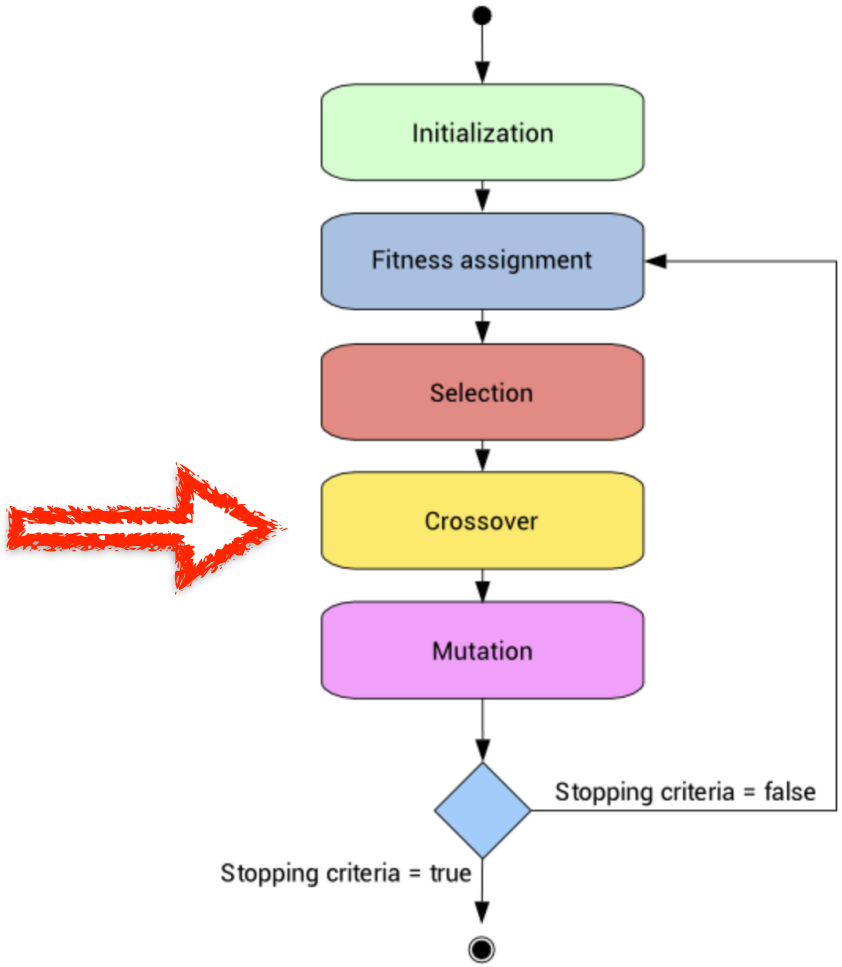
Offen im Denken

- randomly select states with respect to their selection probability
- selected states will be parents for the new generation of the population
- Parameter: **selection rate R** = **what portion of population can survive?**

24748552	→	24	→ 31%
32752411	→	23	→ 29%
24415124	→	20	→ 26%
32543213	→	11	→ 14%

32752411
24748552

Genetic algorithm

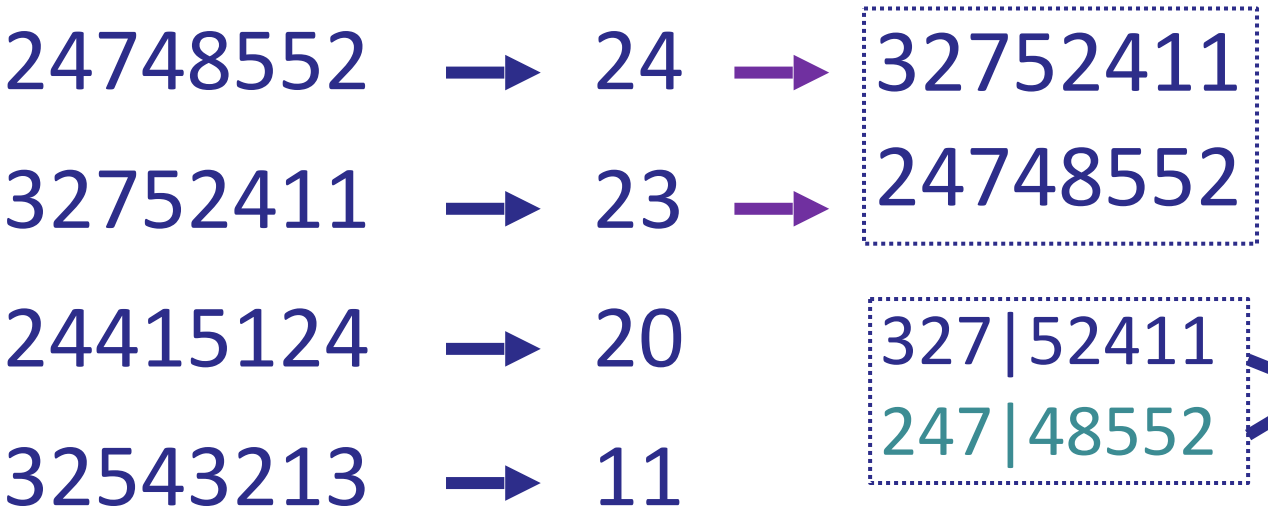


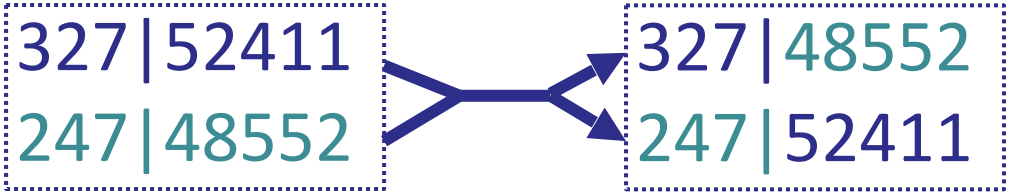
Cross-over

UNIVERSITÄT DUISBURG ESSEN

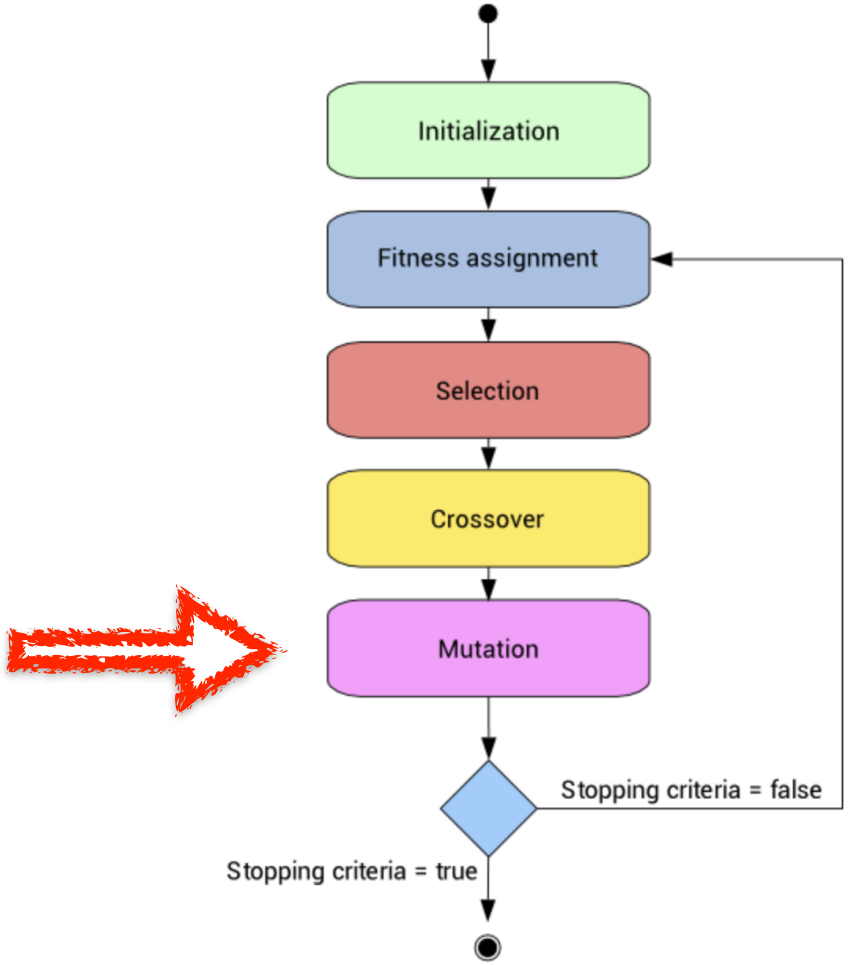
Offen im Denken

- take two parent states
- randomly determine the cross-over point
- cut parents at cross-over point
- recombine the pieces





Genetic algorithm

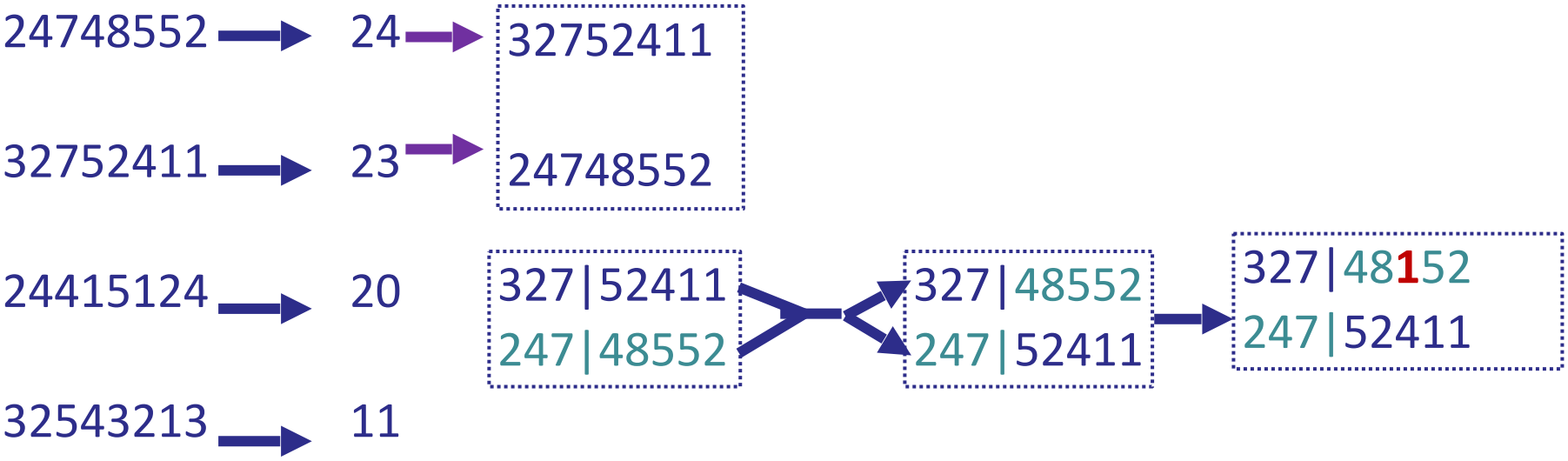


Mutation

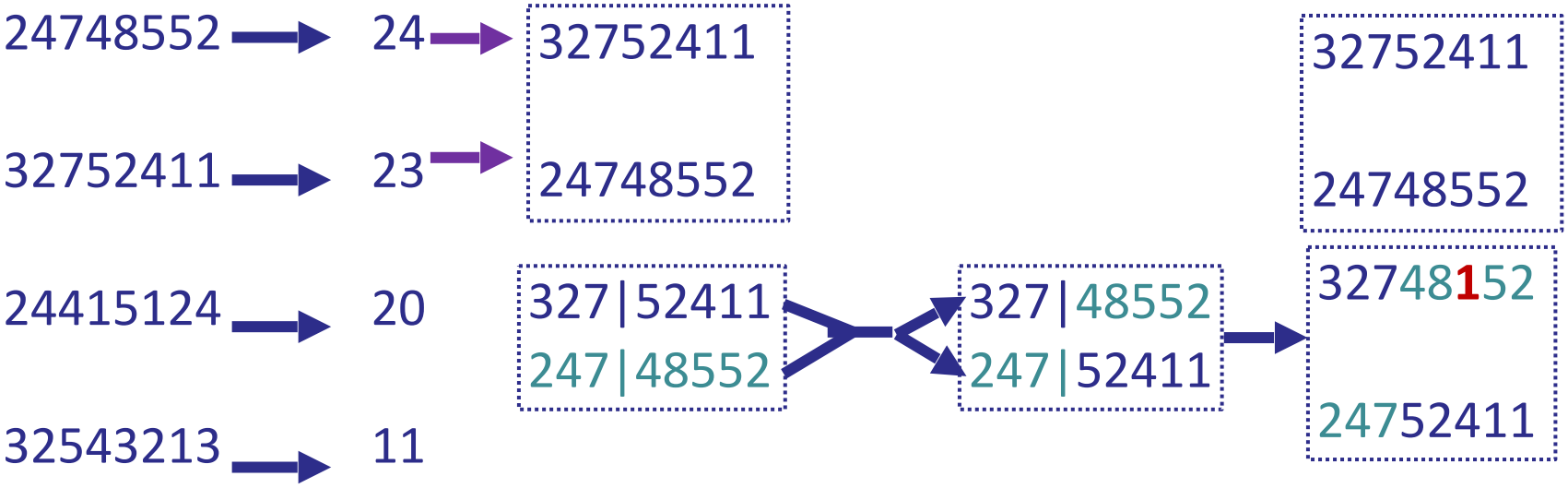
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Random change of n elements in the output states of cross over
  - corresponds loosely to random walks
  - **Parameter: Mutation rate μ: what portion of elements in states should mutate?**



New population



Genetic algorithms

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Properties

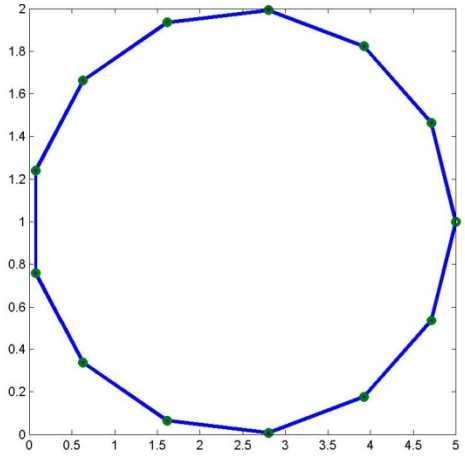
- **easy to implement** and **to explain** general **optimization algorithm**
- **performs well**
  - **unclear under which conditions** they work well
  - other randomized algorithms perform equally well (or better)

- **Applied to optimization problems:**
  - circuit layout,
  - traveling salesman problem (TSP),
  - data clustering and mining,
  - image processing,
  - medical science

Example: Traveling salesperson problem

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

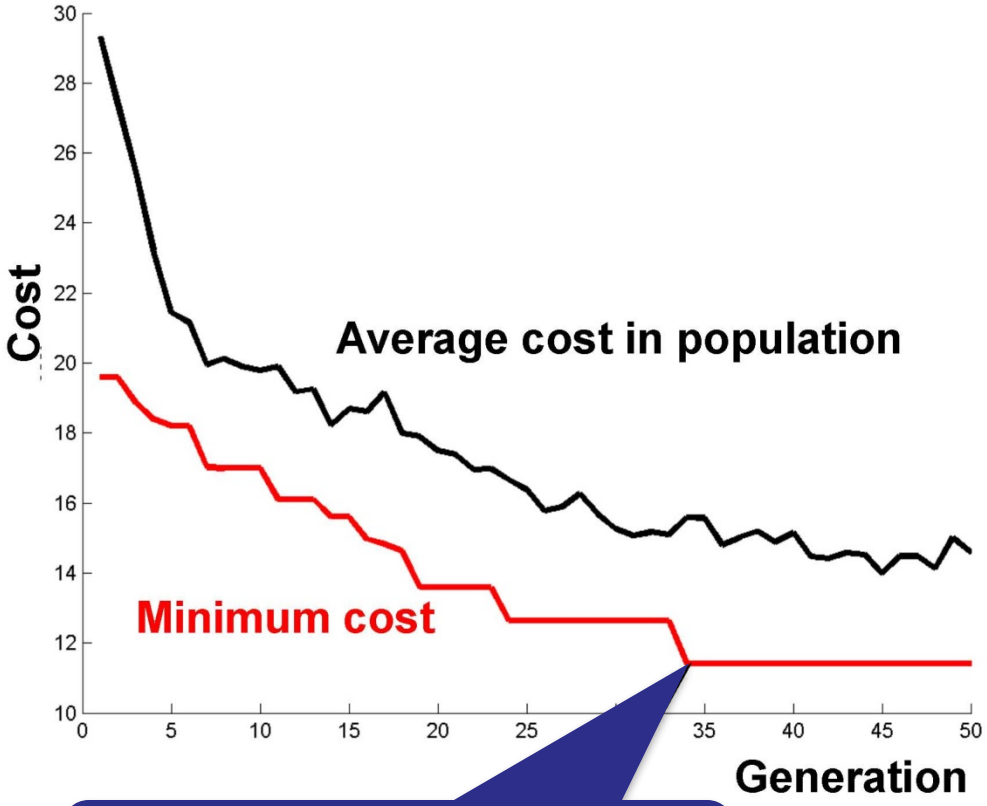


N = 13 cities

P = 100 states in population

μ = 4% mutation rate

R = 50% selection rate



Local search in continuous spaces

UNIVERSITÄT DUISBURG ESSEN

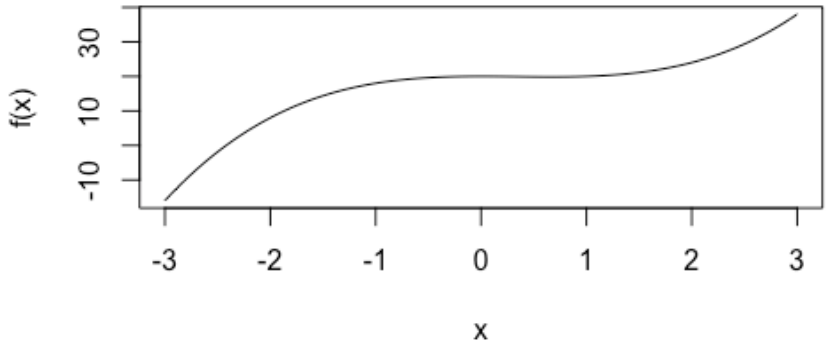
Offen im Denken

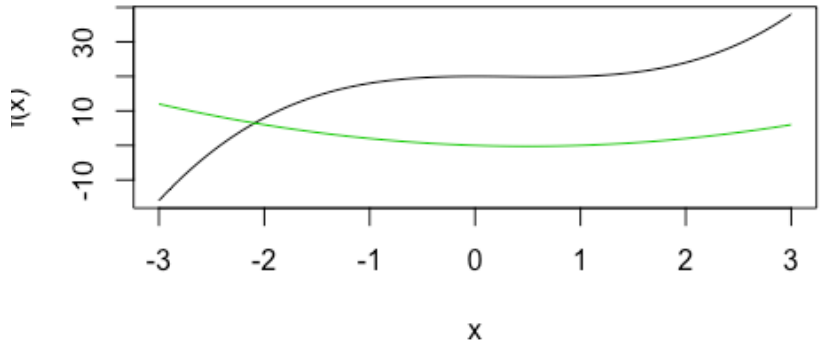
In many real-world problems, the **state space is continuous.**

How to define neighboring states?

- **Discretize** the state space
  - e.g. assume only n different positions of a steering wheel or a gas pedal

- **Gradient Ascent**
  - **hill-climbing** using **the gradient of the objective function** f.
  - What is the **gradient**? **Vector of partial derivations.**





Summary

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Local search: only focuses on the final configuration (not on the path)
- Methods
  - Hill climbing
  - Beam search
  - Simulated annealing
  - Genetic algorithm

→ find good solutions for many practical problems

→ require a careful representation of the problem

  - State representation
  - Evaluation function

Readings

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Mandatory

- Russell & Norvig, Section 4: *Beyond Classical Search*
  - 4.1 *Local search algorithms and optimization problems*, p. 120-129

Optional

- 4.2 and 4.3
- Rest of chapter 4