COMPSCI 761: ADVANCED TOPICS IN ARTIFICIAL INTELLIGENCE

STOCHASTIC SEARCH

Anna Trofimova, August 2022

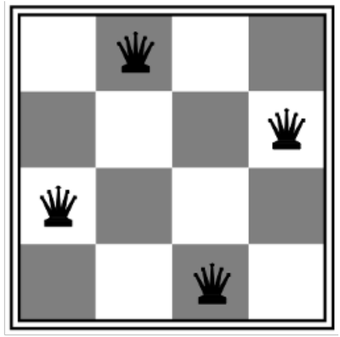
RECAP: SYSTEMATIC VS LOCAL SEARCH

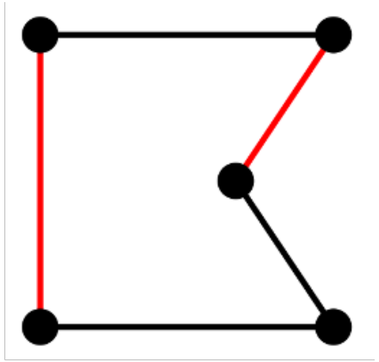
- Systematic search strategies
    - Frontier maintains all unexpanded successors of expanded nodes.
    - Traverse the search space of a problem instance in a systematic manner.
    - Guarantee completeness, i.e., that eventually either a solution is found, or determine that the solution does not exist.

- Local search strategies (also called Iterative Improvement):
    - Frontier maintains some unexpanded successors of expanded nodes.
    - Start at some state and “move” from present location(s) to neighbouring location(s). The moves are determined by the present location(s).
    - *Do not guaranteed* completeness.

RECAP: LOCAL SEARCH ALGORITHMS

- In many optimisation problems, ***path*** is irrelevant; the goal state ***is*** the solution
- Then state space = set of “complete” configurations;
    - find ***configuration satisfying constraints***, e.g., n-queens problem; or, find ***optimal configuration***, e.g., travelling salesperson problem





- In such cases, can use ***iterative improvement*** algorithms: keep a single “current” state, try to improve it
- Constant space, suitable for online as well as offline search

RECAP: HILL-CLIMBING

"*Like climbing Everest in thick fog with amnesia*"

- Simple, general idea:
    - Start wherever
    - Repeat: move to the best neighbouring state
    - If no neighbours better than current, quit

.

TODAY

- Stochastic Local Search
- Random-restart Hill-Climbing
- First Choice Hill-Climbing
- Stochastic Hill-Climbing
- Random Walk Hill-Climbing
- Probabilistic Hill-Climbing

STOCHASTIC LOCAL SEARCH

- Stochastic local search (SLS) are local search strategies where randomisation plays a prominent role:
    - **Initialisation**: Random start
    - **Tie breaking:** Random tie breaking
    - **Step function:** e.g. random ordering of successors (first-choice hill-climbing), random tabu tenure.
- The local search strategies that we introduced earlier can all be extended by stochastic local search.

.

RANDOM-RESTART HILL CLIMBING

- series of hill-climbing searches
    - (from randomly generated initial states)
- overcomes local maxima
    - trivially complete
- Hill-climbing with random restarts runs a sequence of Hill-climbing searches:
    1. Run hill-climbing from a random starting candidate
    2. Repeat Step (1)

HILL-CLIMBING WITH RANDOM RESTARTS

- If at first you don’t succeed, try, try again!
- Different variations
    - For each restart: run until termination vs. run for a fixed time
    - Run a fixed number of restarts or run indefinitely
- Analysis
    - Say each search has probability p of success
        - E.g., for 8-queens, p = 0.14 with no sideways moves
    - Expected number of restarts?
    - Expected number of steps taken?

CALCULATING THE EXPECTED NUMBER OF RUNS

- Since each try of local search starts at random.
- Whether a try is successful or not is a Bernoulli random variable.
- Suppose a single try of greedy descent has a success rate of p., then the expected number of tries needed to find the optimal solution is $1/p$.

.

8-QUEENS WITH RANDOM RESTARTS

- p is probability of success

- Number of restarts is 1/p
    - 8 queens, no sideways, p=0.14, so ~7 restarts
    - 8 queens, sideways, p=0.94, so ~1.06 restarts

- Number of steps
    - (1 x cost of success) + ((1/p - 1) x cost of failure)
    - 8 queens, no sideways, (1 x 4) + ((.86/.14) x 3) ~ 22 steps
    - 8 queens, sideways, (1x21) + ((0.06/0.94) x 64) ~ 25 steps

- For 3 million queens, random restart finds solutions in under a minute

.

FIRST CHOICE HILL-CLIMBING IDEA

- Implementation of stochastic hill-climbing
- Generating successors randomly until it finds one which is uphill
    - Very good when a state has many successors
    - Don’t waste time generating them all

.

STOCHASTIC HILL-CLIMBING

- Stochastic hill-climbing
    - Random selection among the uphill moves.
    - The selection probability can vary with the steepness of the uphill move.
- Stochastic Hill Climbing chooses a random better state from all better states in the neighbours
    - First-Choice Hill Climbing chooses the first better state from randomly generated neighbours.
- Can still get stuck in local maxima!

RANDOM WALK HILL-CLIMBING

Random Walk Hill-Climbing works by assuming that adding randomisation may avoid the search getting stuck in a local optimal.

- Pick a parameter (walk probability) $p \in [0, 1]$
- At every step,
  - With probability p, make an uninformed random walk, i.e., choose a successor uniformly at random and move there.
  - With probability $1 - p$, make a greedy choice: choose the successor that has the least loss.

RANDOM WALK HILL-CLIMBING

- If the set of states, S, is finite and
    - every state is reachable from every other state by the step function

- Then Random Walk Hill-Climbing is PAC.

- A search strategy is probabilistic approximately complete (PAC) if the probability that a try fails to find an optimal solution can be made arbitrarily small when the search runs for sufficiently long.

.

RANDOM WALK & RANDOM RESTART HILL-CLIMBING

At each step do one of the three

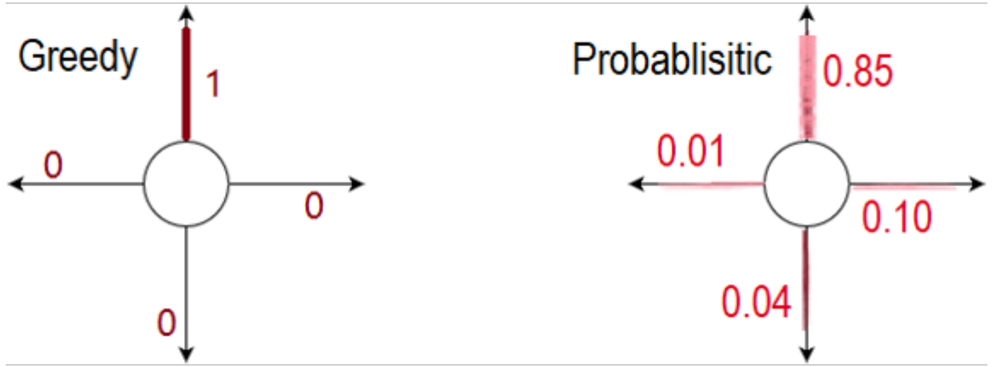
- Greedy: move to the neighbour with largest value
- Random Walk: move to a random neighbor
- Random Restart: Resample a new current state

.

PROBABILISTIC HILL-CLIMBING

Probabilistic hill-climbing allows worsening search steps with a probability that depends on the respective deterioration in evaluation function value.

“The worse a step is, the less likely it would be performed”

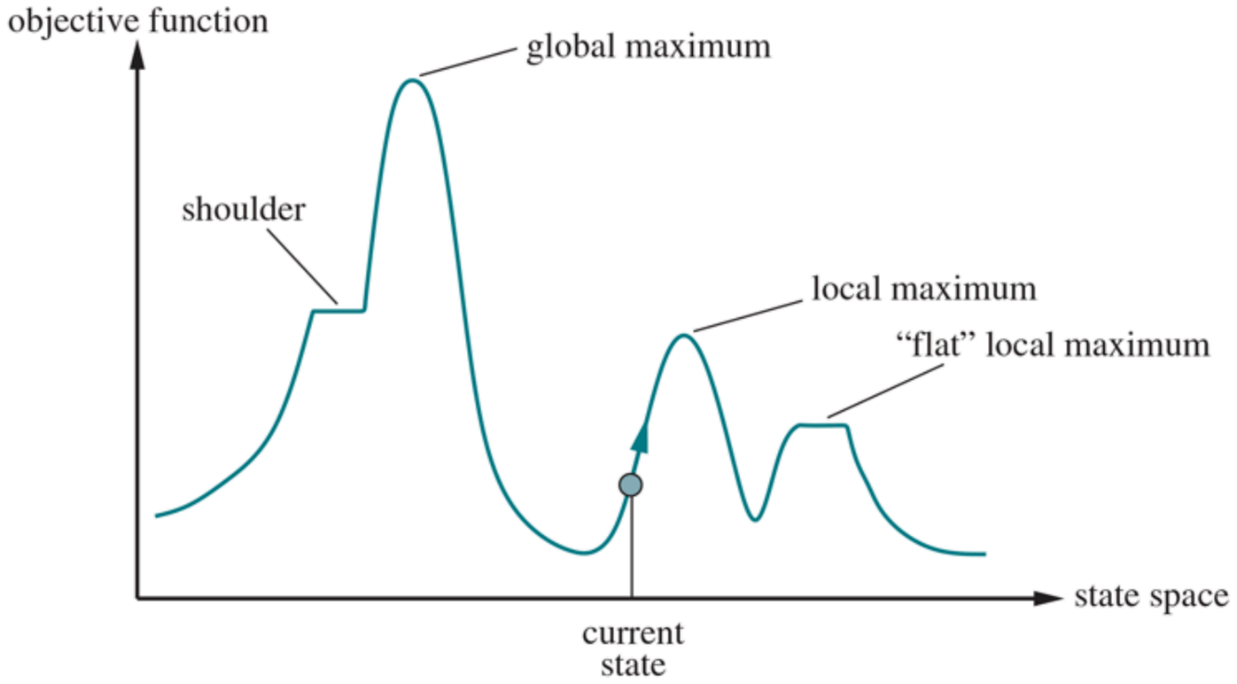


PROBABILISTIC HILL-CLIMBING: MAIN IDEA

Choosing the successor that has the biggest drop in loss may lead to local optima that are not global optima.

To get out of local optima, it is necessary to choose other successors.

We are still favouring successors that bring faster decrease on the loss.



ANNEALING

Annealing is a metallurgical process where molten metals are slowly cooled to allow them to reach a low energy state, making them stronger.

At high temperature, the thermodynamic movement tends to be more random.

At low temperature, there is little random movements.

The annealing process would start by keeping the metal above its recrystallisation temperature for a while before letting it cool down.

.

SIMULATED ANNEALING

- Resembles the annealing process used to cool metals slowly to reach an ordered (low-energy) state
- Basic idea:
  - Allow “bad” moves occasionally, depending on “temperature”
  - High temperature => more bad moves allowed, shake the system out of its local minimum
  - Gradually reduce temperature according to some schedule

AN EXAMPLE

A physical analogy:

Imagine letting a ball roll downhill on an uneven surface The ball may get stuck at local minima.

Now imagine shaking the surface, while the ball rolls, gradually reducing the amount of shaking.

SIMULATED ANNEALING ALGORITHM

```
function SIMULATED-ANNEALING(problem,schedule) returns a  state
current ← problem.initial-state
for t = 1 to ∞ do
        T ←schedule(t)
        if T = 0 then return current
        next ← a randomly selected successor of current
        ΔE ← next.value – current.value
        if ΔE > 0 then current ← next
                    else current ← next only with probability e^(ΔE/T)
```

T temperature

ΔE badness of move

TEMPERATURE T

- high T: probability of “locally bad” move is higher
- low T: probability of “locally bad” move is lower

- typically, T is decreased as the algorithm runs longer
    - i.e., there is a “temperature schedule”

.

TYPICAL ANNEALING SCHEDULE

- Usually use a decaying exponential
- Axis values scaled to fit problem characteristics

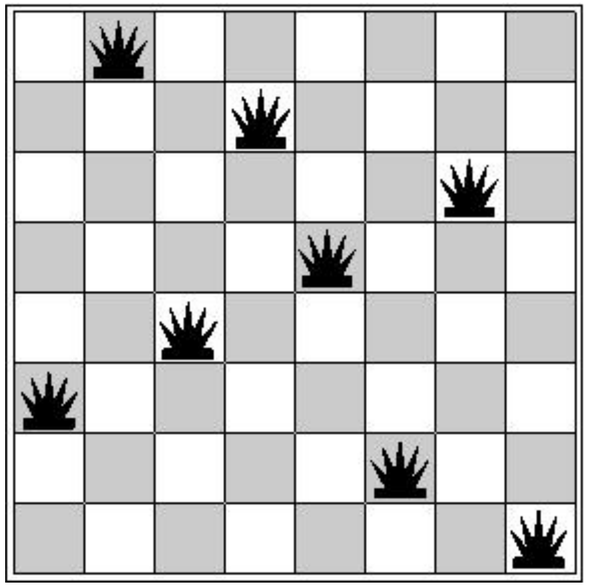
PROBABILITY OF ACCEPTING WORSE SUCCESSOR

- Decreases as temperature T decreases
- Probability Increases as △E decreases
- Sometimes, step size also decreases with T

△E is always negative!

exp(△***E*** / T)		Temperature T	
		High	**Low**
△***E***	**High**	Medium	Low
	Low	High	Medium

SIMULATED ANNEALING EXAMPLE: 8-QUEENS



- Is this a solution?
- What is the value of h?
- Hill-climbing with Min Conflicts would get stuck in this state
- Simulated Annealing, however, can accept such a move with probability exp(-(h1-h0)/T), thus bumping the system out of this local optimum and allowing it to continue the search for a global optimum

SIMULATED ANNEALING

- Theoretical guarantee:
  - Stationary distribution (Boltzmann): $p(x) \propto e^{\frac{E(x)}{kT}}$
  - where *k* is Boltzmann's constant, and *T* is the thermodynamic temperature.
  - If *T* decreased slowly enough, will converge to optimal state!

SIMULATED ANNEALING

Converge to optimal state sounds like magic, but reality is reality:

- The more downhill steps you need to escape a local optimum, the less likely you are to ever make them all in a row
- “Slowly enough” may mean exponentially slowly
- Random restart hill-climbing also converges to optimal state…

.

PROPERTIES OF SIMULATED ANNEALING

- Unfortunately, convergence might take a VERY VERY long time
    - So, ultimately this is a very weak claim

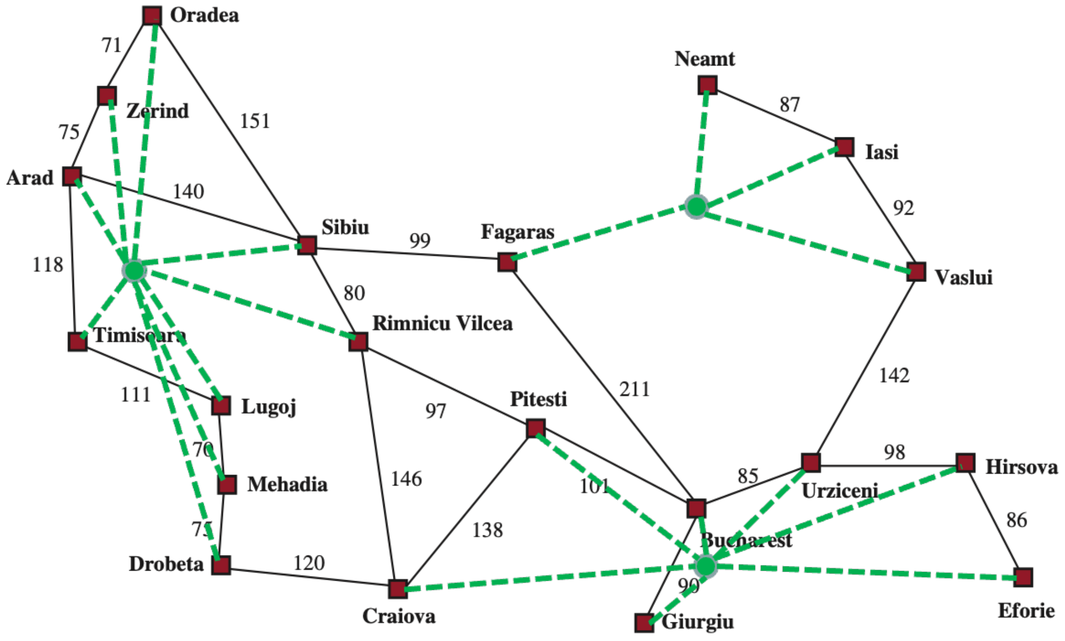
- Often works very well in practice
    - But usually VERY VERY slow
        - slowness comes about because T must be decreased very gradually to retain optimality
- Simulated annealing and its relatives are a key workhorse in VLSI layout and other optimal configuration problems

.

LOCAL SEARCH IN CONTINUOUS SPACES

EXAMPLE: SITING AIRPORTS IN ROMANIA

Place 3 airports to minimise the sum of squared distances from each city to its nearest airport.



Airport locations

$\mathbf{x} = (x_1,y_1), (x_2,y_2), (x_3,y_3)$

City locations (x_c,y_c)

C_a = cities closest to airport a

Objective: minimize

$f(\mathbf{x}) = \Sigma_a \Sigma_{c \in Ca} (x_a - x_c)^2 + (y_a - y_c)^2$

OPTIMISATION OF CONTINUOUS FUNCTIONS

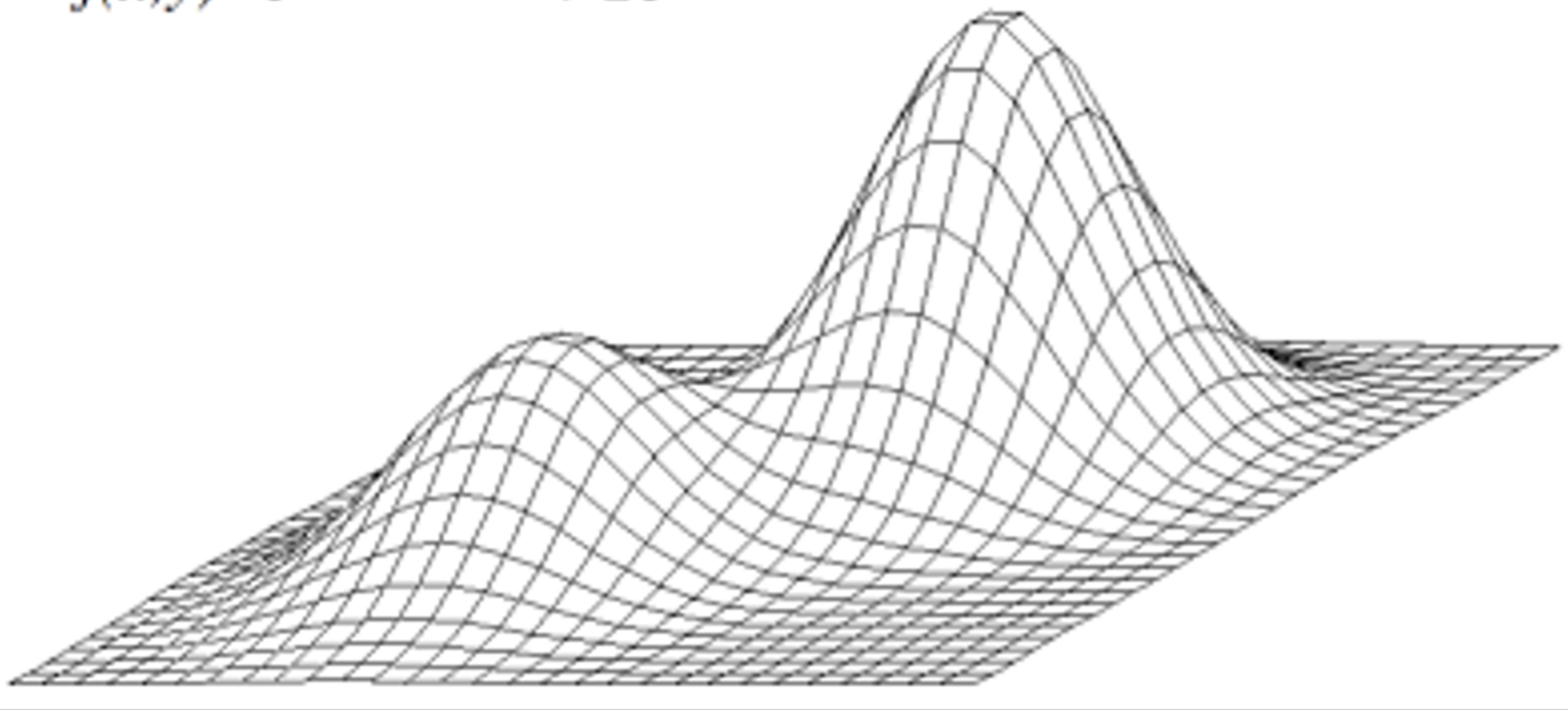
- Discretisation
    - use hill-climbing
- Gradient descent
    - make a move in the direction of the gradient
    - gradients: closed form or empirical

HANDLING A CONTINUOUS STATE/ACTION SPACE

1. Discretise it!
   - Define a grid with increment d, use any of the discrete algorithms
2. Choose random perturbations to the state
   a. First-choice hill-climbing: keep trying until something improves the state
   b. Simulated annealing
3. Compute gradient of $f(\mathbf{x})$ analytically

CONTINUOUS SPACE

$$f(x,y)=e^{-(x^2+y^2)} + 2e^{-((x-1.7)^2+(y-1.7)^2)}$$



GRADIENT DESCENT

Assume we have a continuous function: $f(x_1,x_2,\ldots,x_N)$ and we want minimise over continuous variables X, X2, .., Xn

Gradient = the most direct direction up-hill in the objective (cost) function, so its negative minimises the cost function.

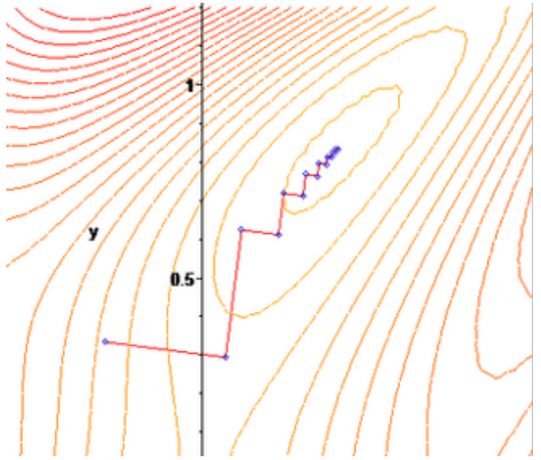
1. Compute the gradients for all i: $df(x_1,x_2,\ldots,x_N) / dx_i$

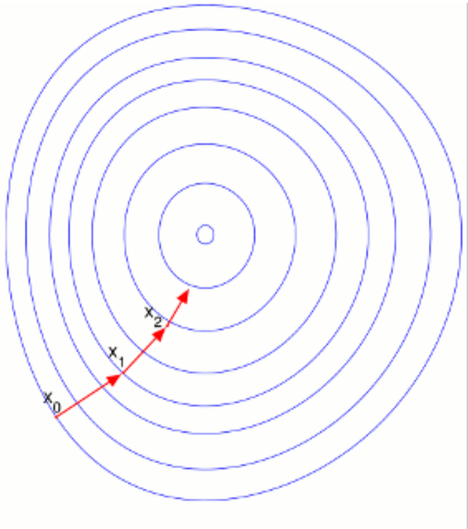
2. Take a small step downhill in the direction of the gradient:

$$x_i \leftarrow x_i - \lambda df(x_1,x_2,\ldots,x_N) / dx_i$$

3. Repeat.

How to select λ?





SUMMARY I

- Many configuration and optimisation problems can be formulated as local search
- General families of algorithms:
    - Local Search (+ continuous optimisation)
    - Stochastic Search (Simulated annealing and other stochastic methods)
    - Population Based search: (next lecture)

Many machine learning algorithms are local searches

.

SUMMARY II - LOCAL SEARCH

- Local Search + Gradient Descent
    - Hill-Climbing
    - Hill-Climbing with Sideways Moves
    - Tabu Search
    - Enforced Hill-Climbing

- Stochastic Local Search
    - First Choice Hill-Climbing
    - Stochastic Hill-Climbing
    - Random Walking Hill-Climbing
    - Random Restart Hill-Climbing
    - Simulated Annealing

- Population Based Search (next time!)

SUMMARY III - STOCHASTIC LOCAL SEARCH

- Stochastic Hill Climbing – chose probabilistically from among better children based on their fitness
- First Choice Hill-Climbing - chose the first better random child
- Random Walking Hill Climbing – chose probabilistically between the best child and a random child
- Random Restart Hill Climbing – do any hill climbing method with randomly restarts to try and get a better result
- Simulated Annealing – first choice hill climbing with a stochastic chance of choosing a worse node. The chance of choosing a worse node reduces over time with the temperature schedule