Prepared by Julian García, based on material by
Arun Konagurthu

Workshop 18
Heuristics: Simulated Annealing

FIT 3139 Computational Modelling and Simulation

MONASH University
Information Technology

heuristic | ˌhjʊ(ə)ˈrɪstɪk |

adjective

enabling a person to discover or learn something for themselves: *a 'hands-on' or interactive heuristic approach to learning*.

- *Computing* proceeding to a solution by trial and error or by rules that are only loosely defined.

noun

a heuristic process or method.

- (**heuristics**) *[usually treated as singular]* the study and use of heuristic techniques.

DERIVATIVES

---

heuristically adverb

ORIGIN

---

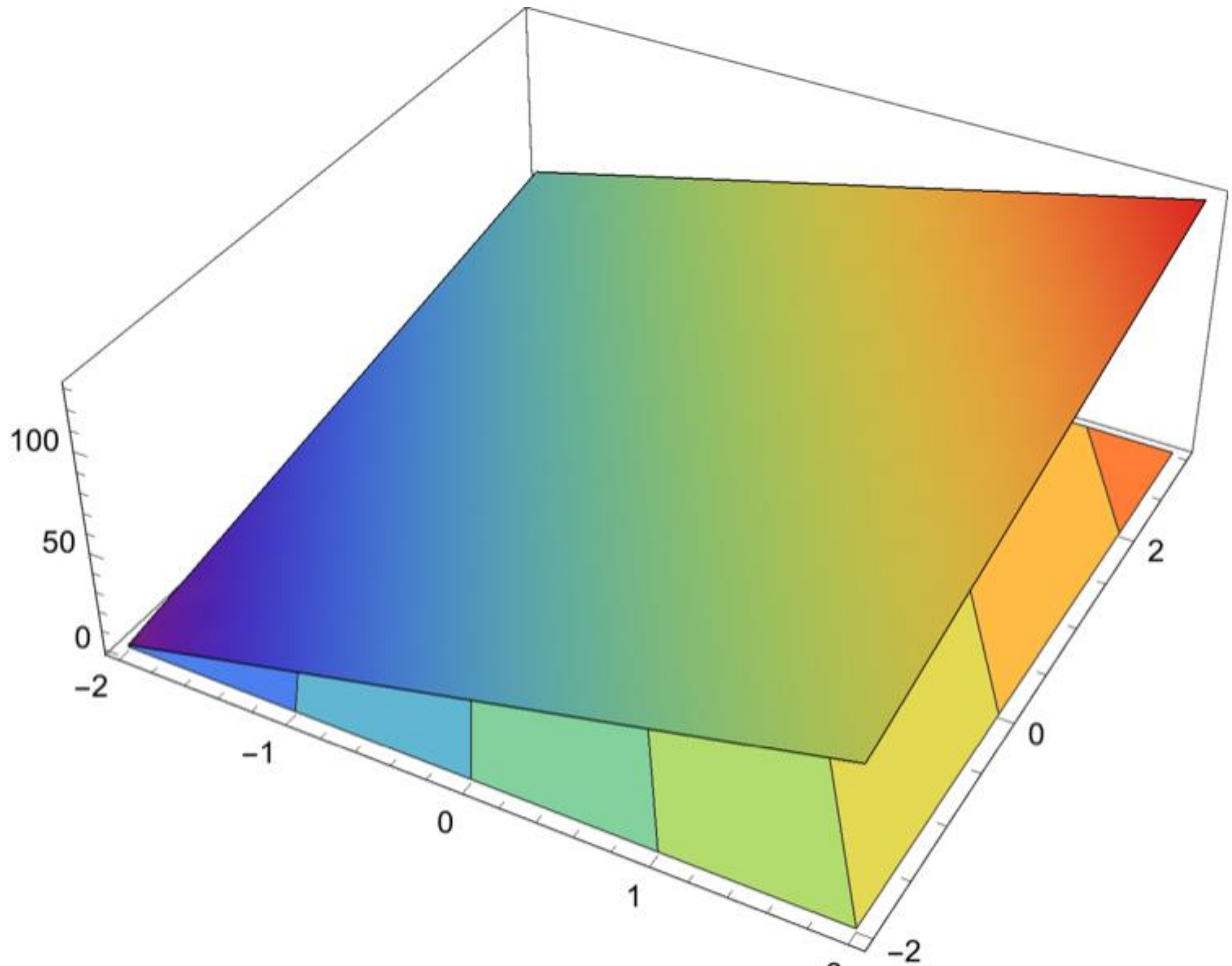
early 19th century: formed irregularly from Greek ***heuriskein*** 'find'.

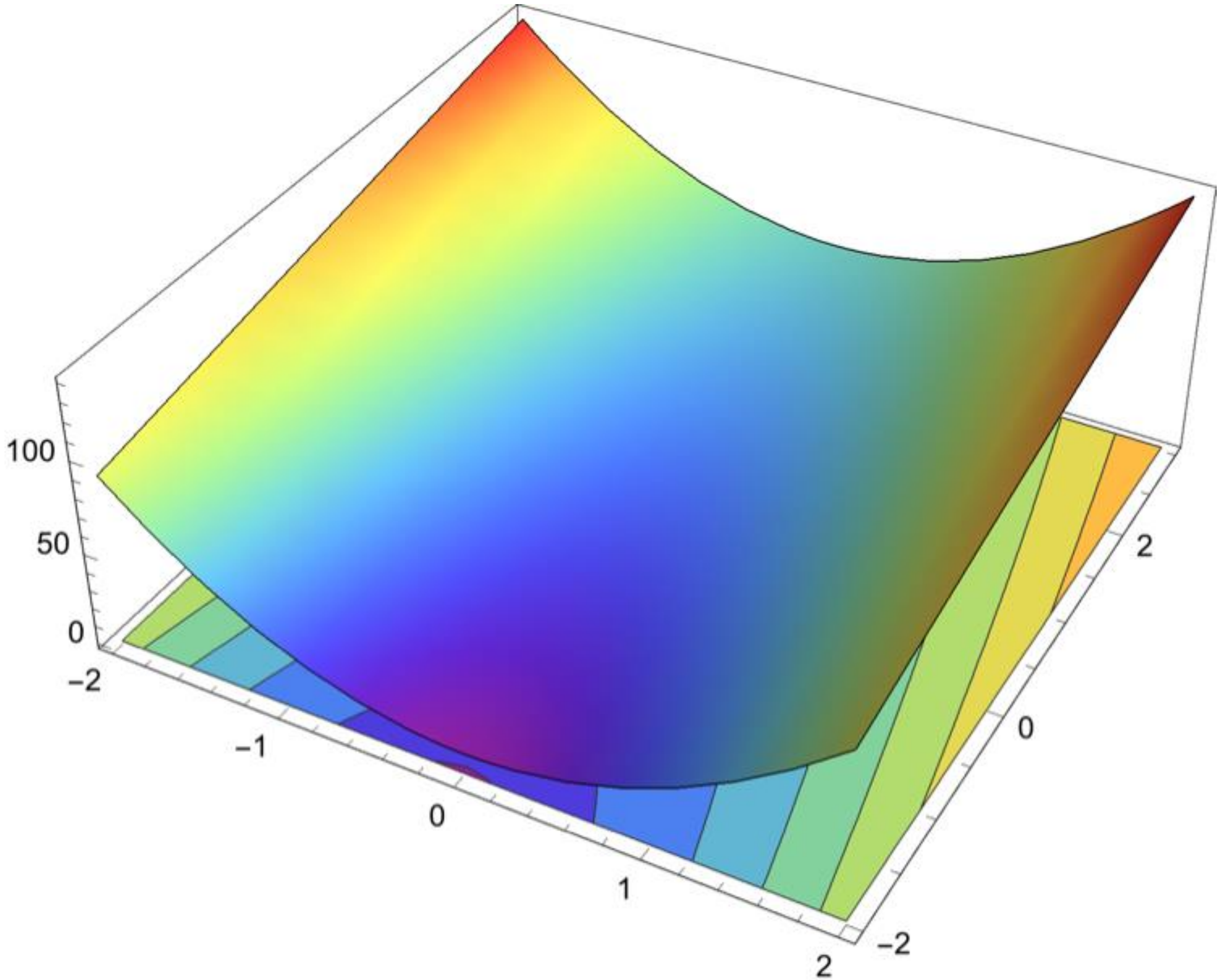
Heuristics:

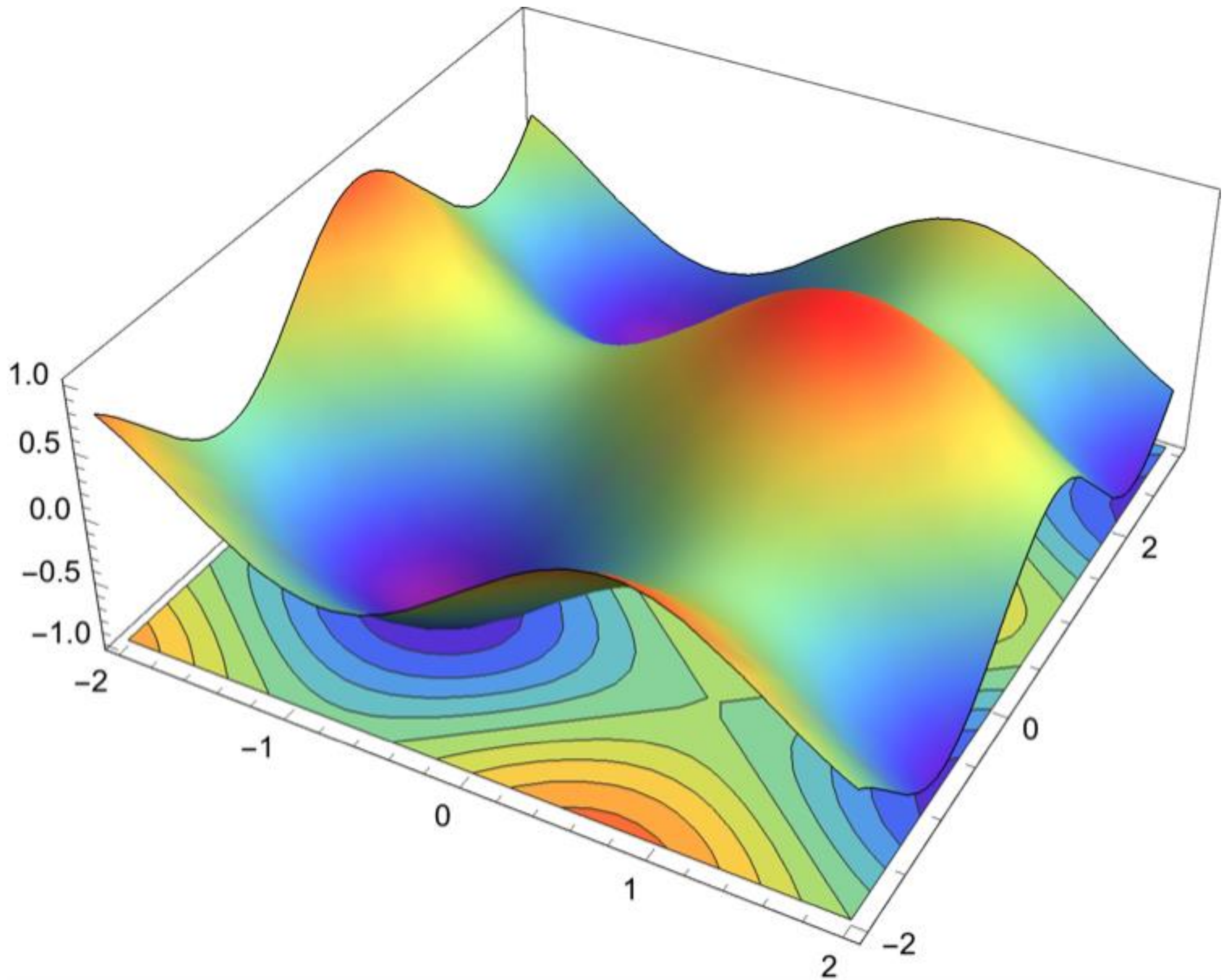
A family of techniques that ignore if solutions can be proven to be correct, but usually produces **a good solution**.

When/Why use Heuristics:

Intractable large and complex problems (non-convexity, non-linearity) that are **difficult to solve optimally**.


100
50
0
−2
−1
0
1
2
−2
0
2


100
50
0
−2
−1
0
1
2
−2
0
2



1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2
−2
0
2

Heuristics:

A family of techniques that ignore if solutions can be proven to be correct, but usually produces **a good solution**.

When/Why use Heuristics:

Intractable large and complex problems (non-convexity, non-linearity) that are **difficult to solve optimally**.

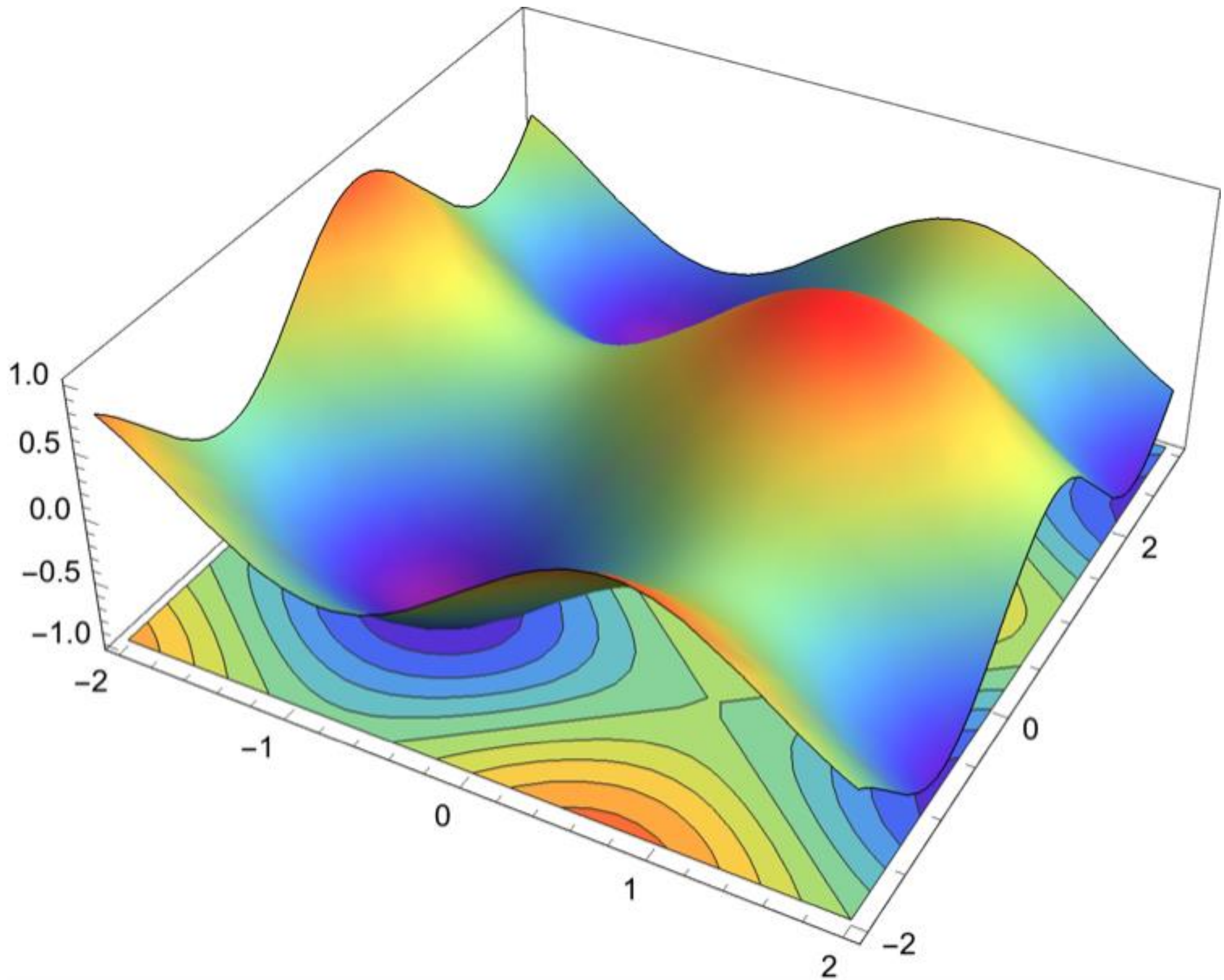
Outline:

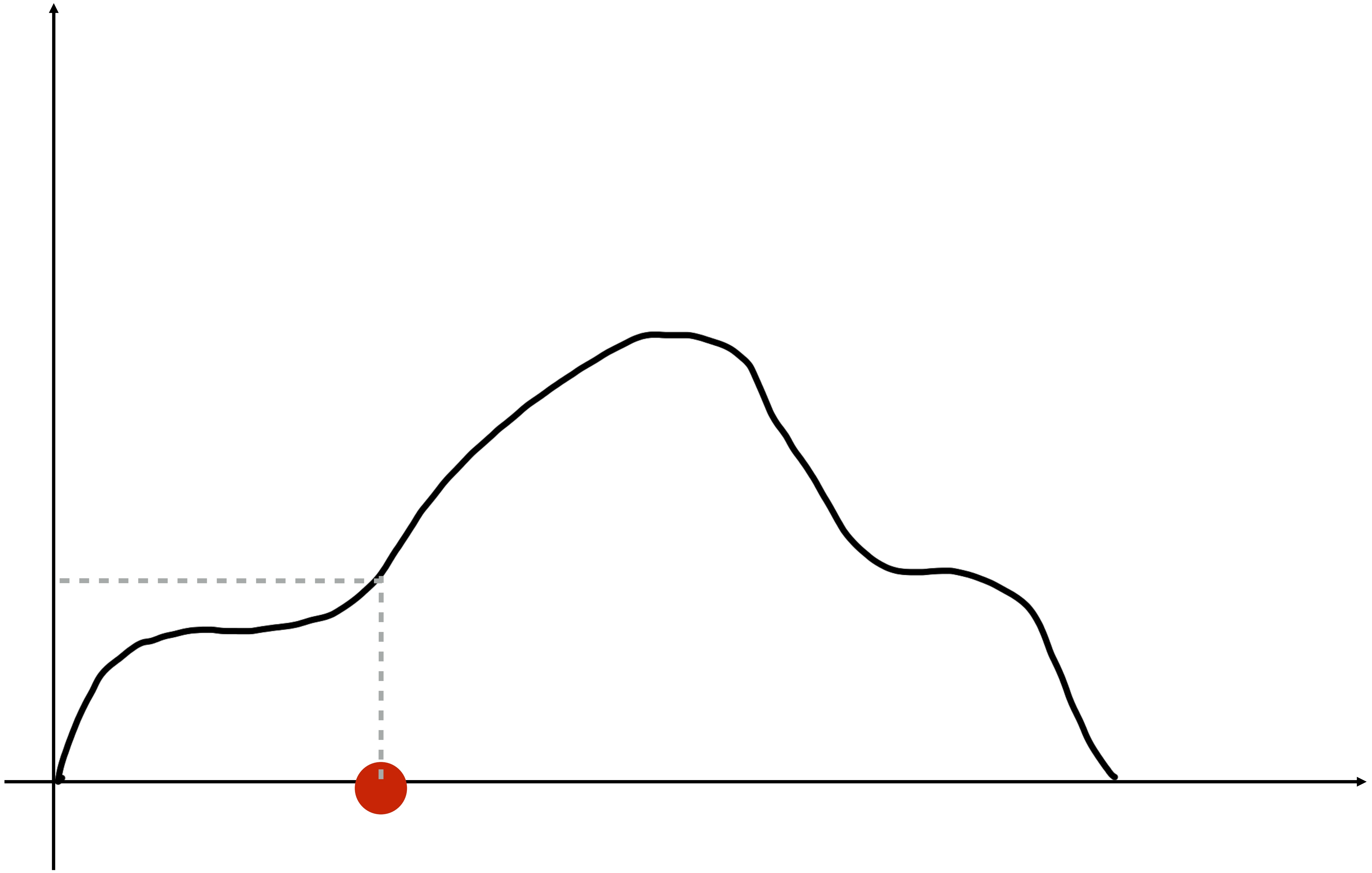
- Simulated Annealing (Metropolis)
- Evolutionary Computation (Genetic Algorithms)

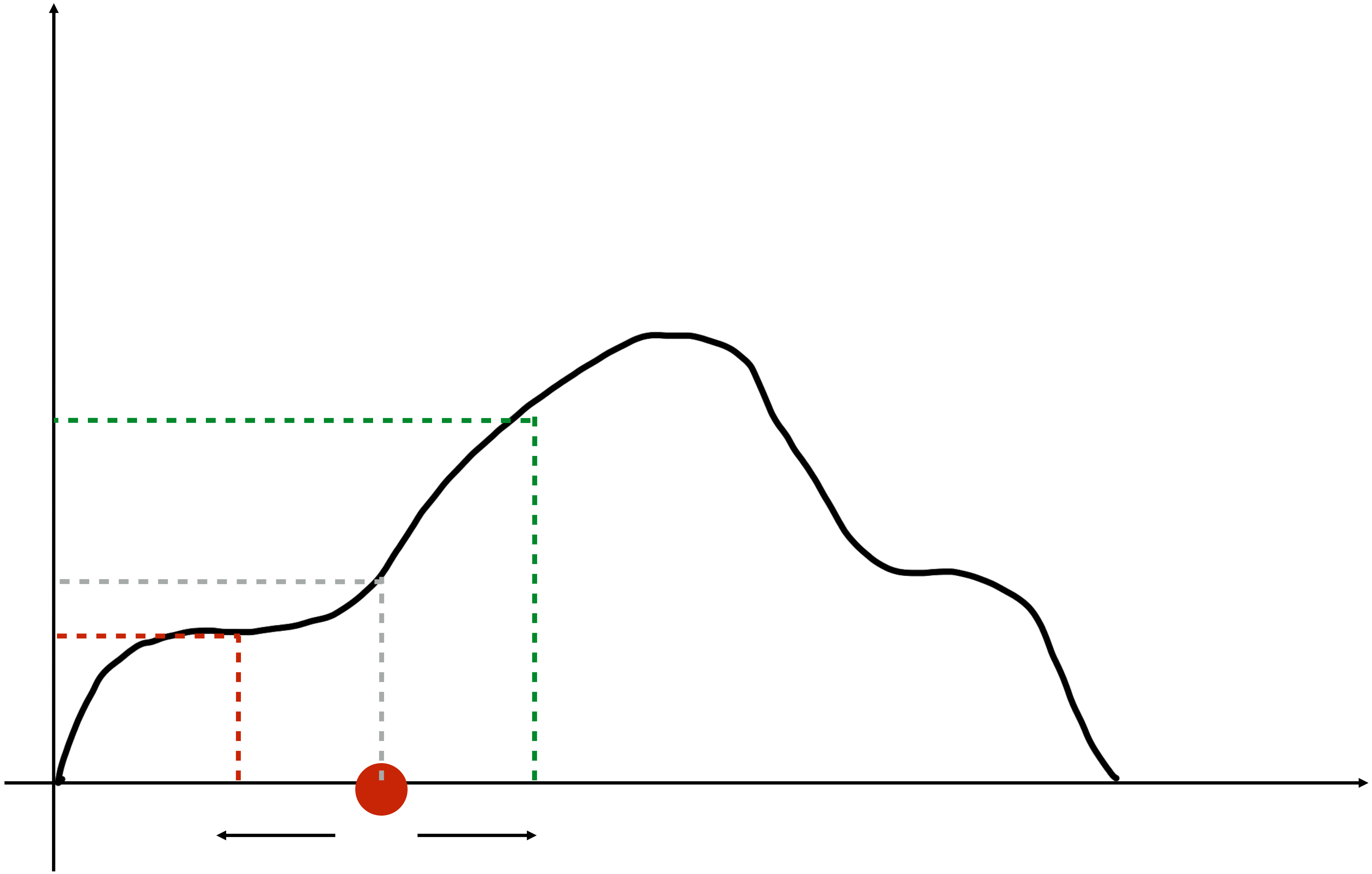
Both techniques are ***Heuristics***

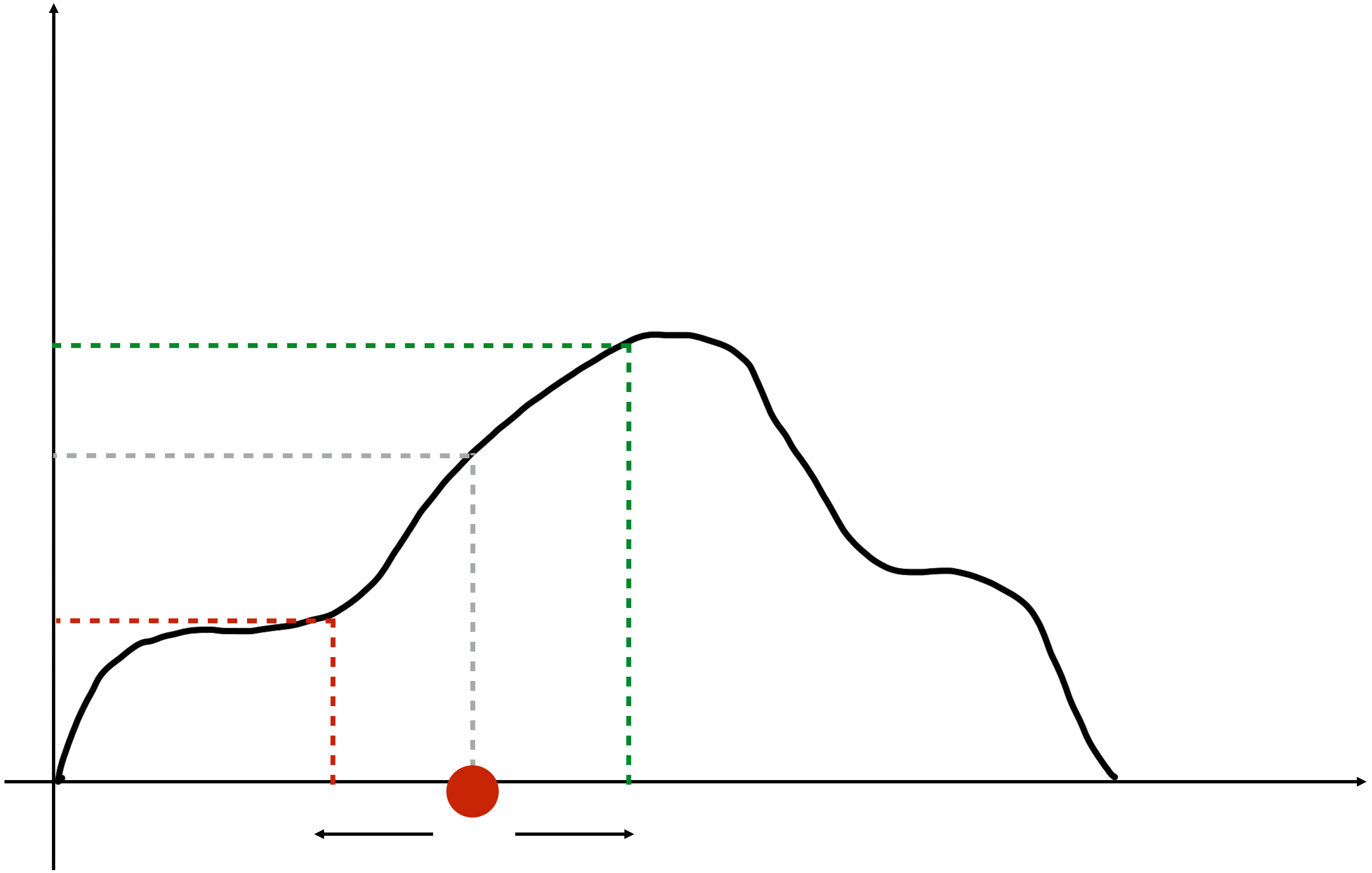
"Natural computation"

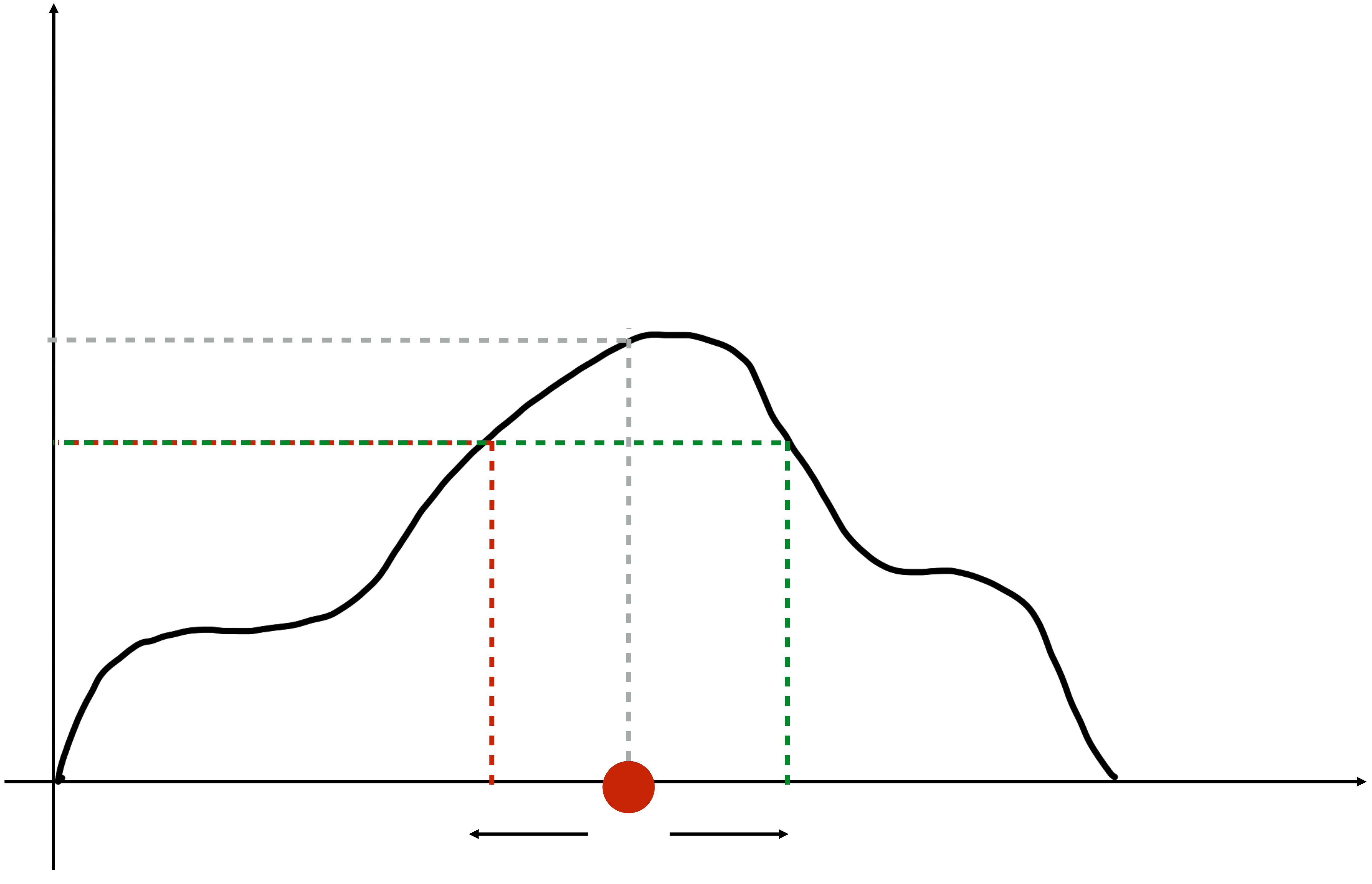
inspired in natural processes.

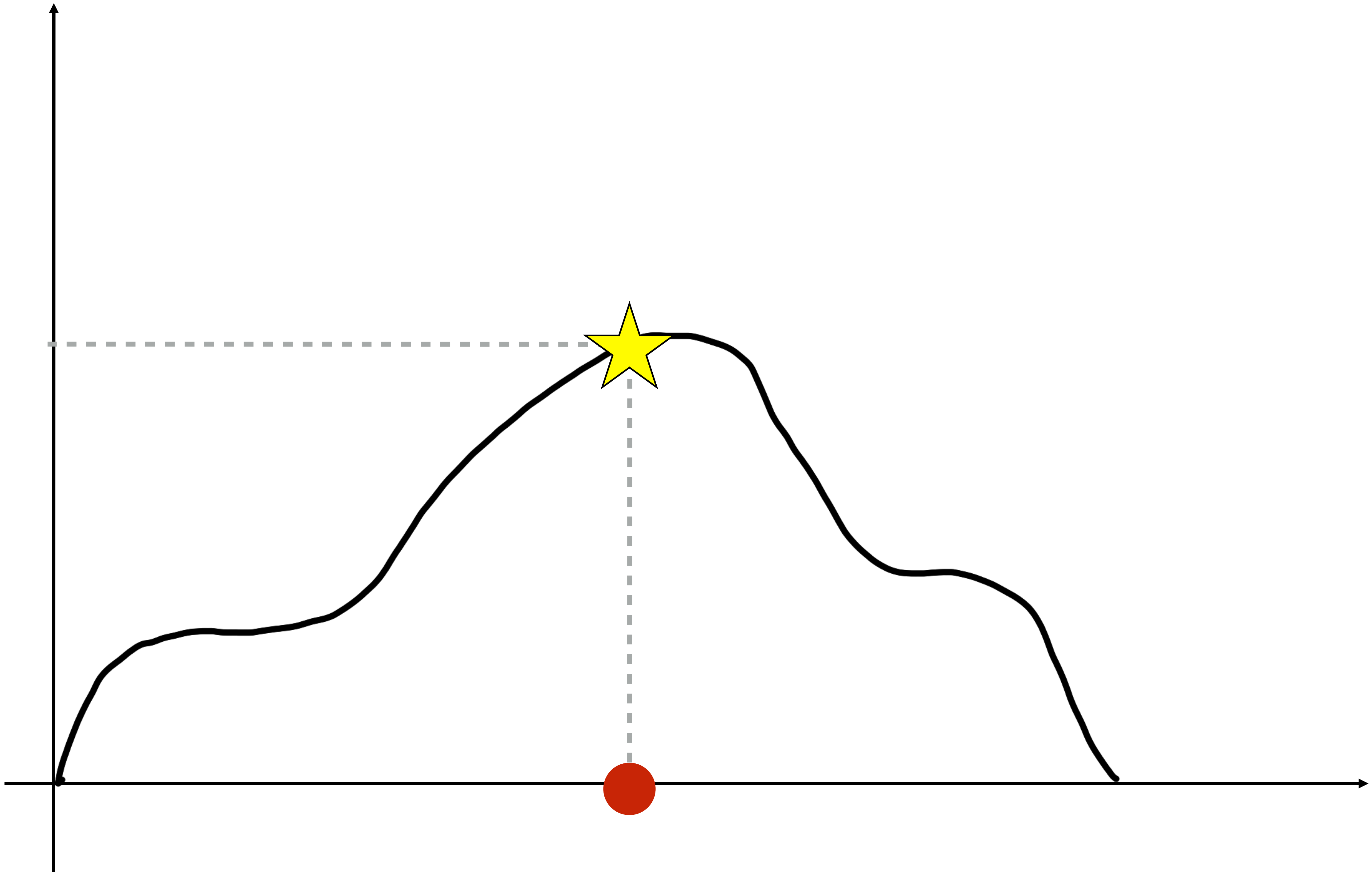

1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2
−2
0
2



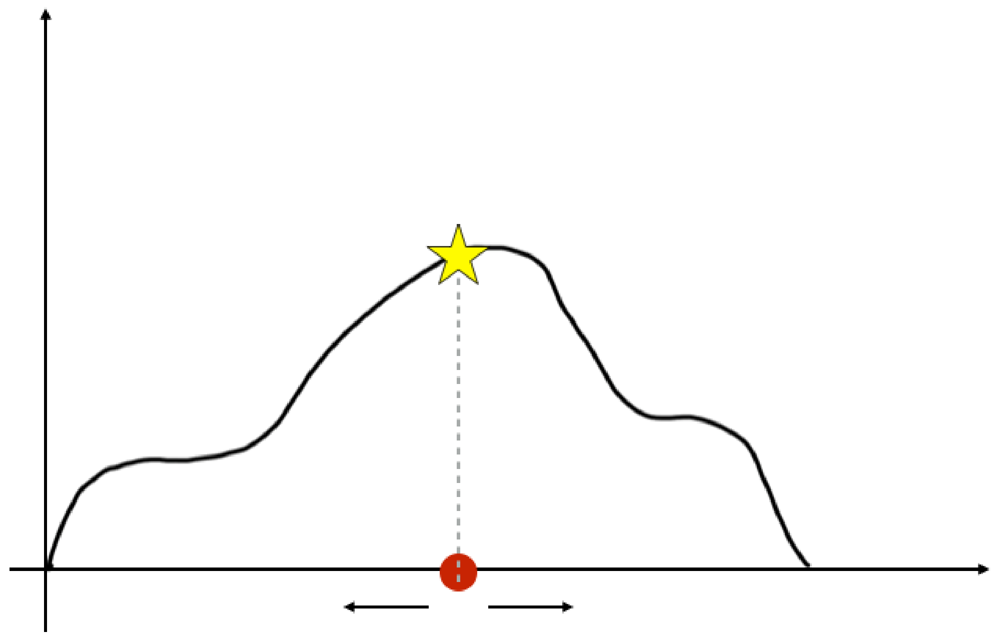








Hill Climbing

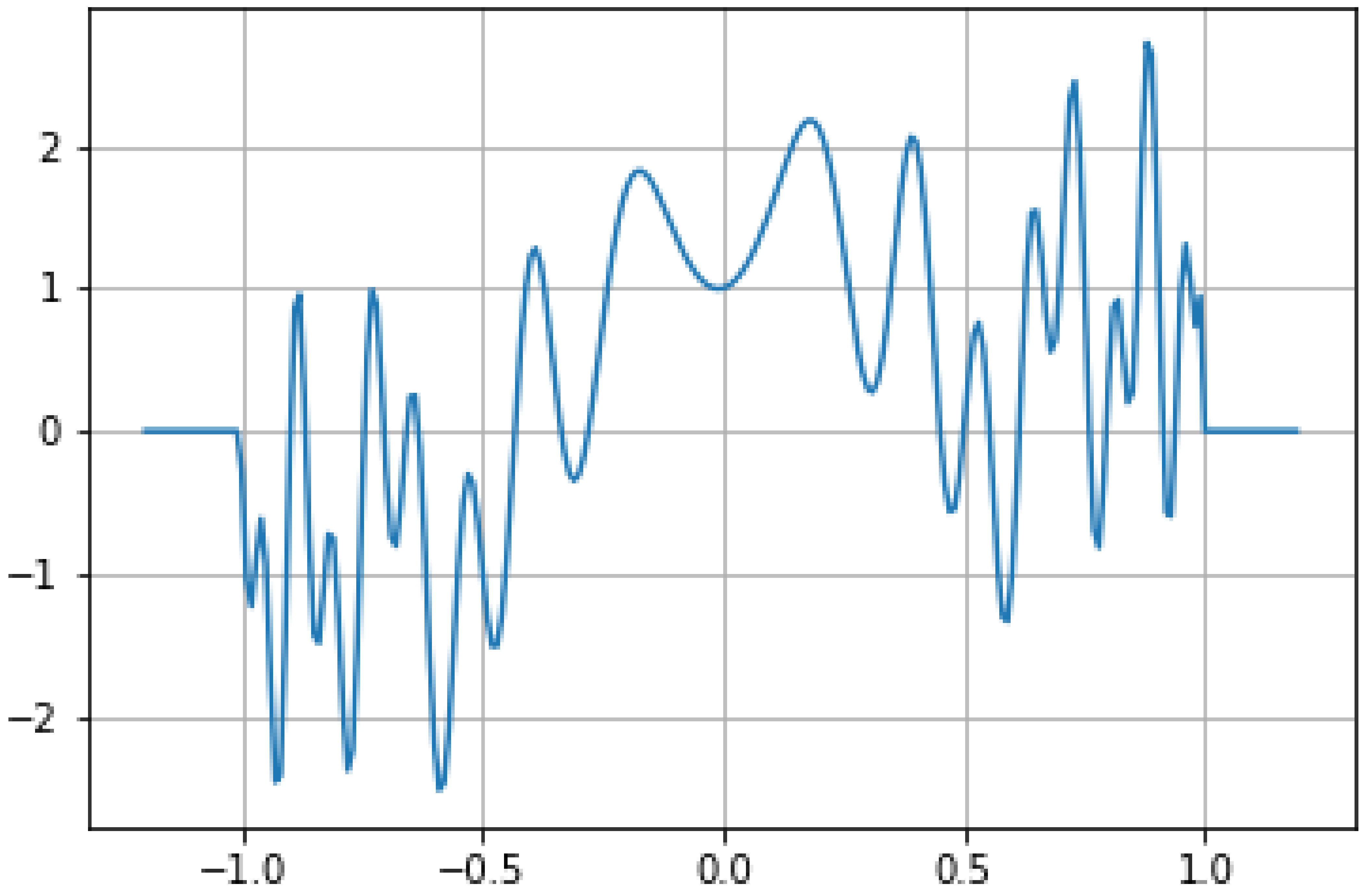


1. start with an arbitrary solution

2. attempt to find a better solution by making a *local* (i.e., incremental) change to the current solution.

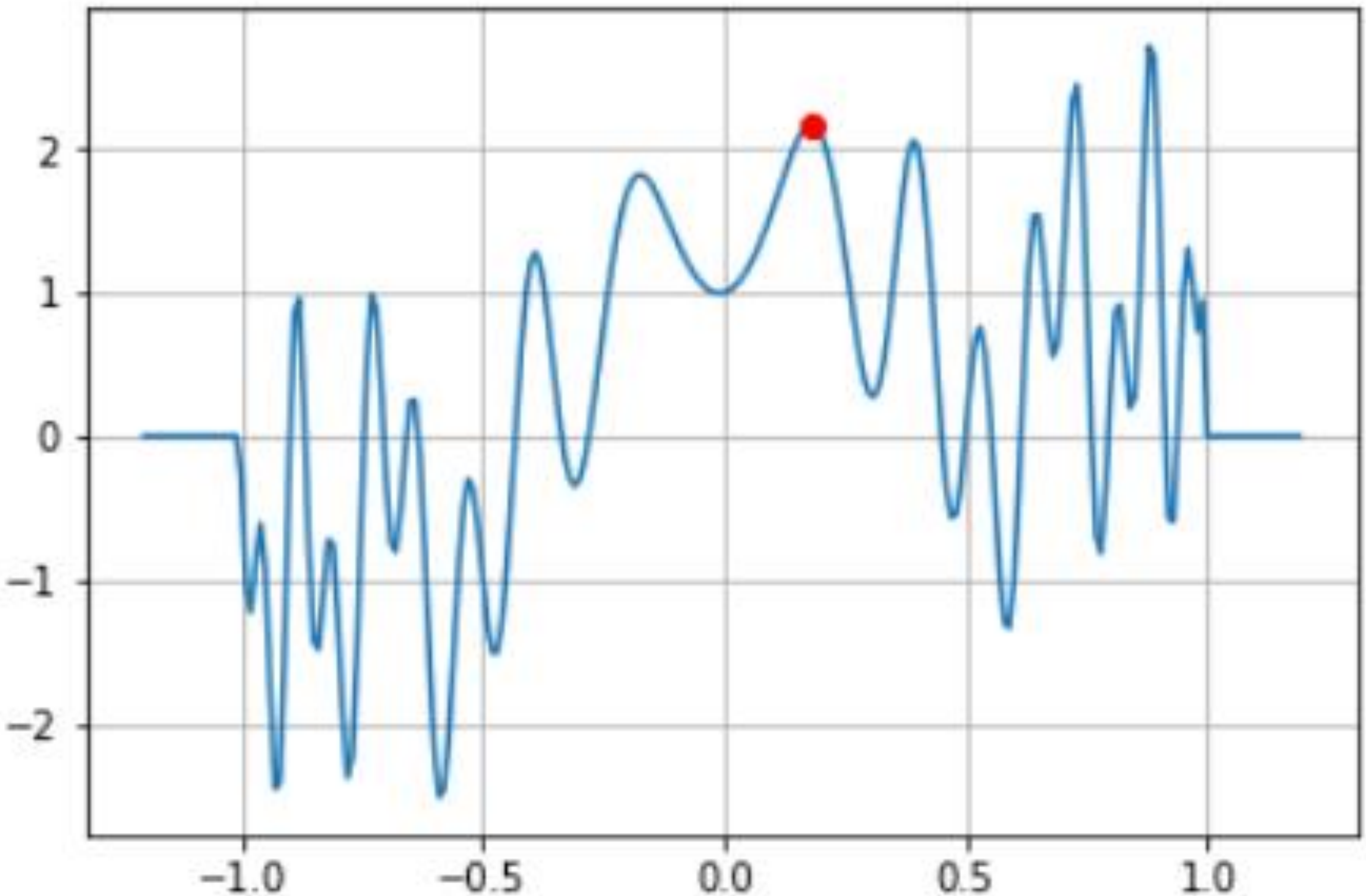
3. continue to make incremental changes until no further improvements can be found

Is there a particular kind of problem where this procedure may be problematic?



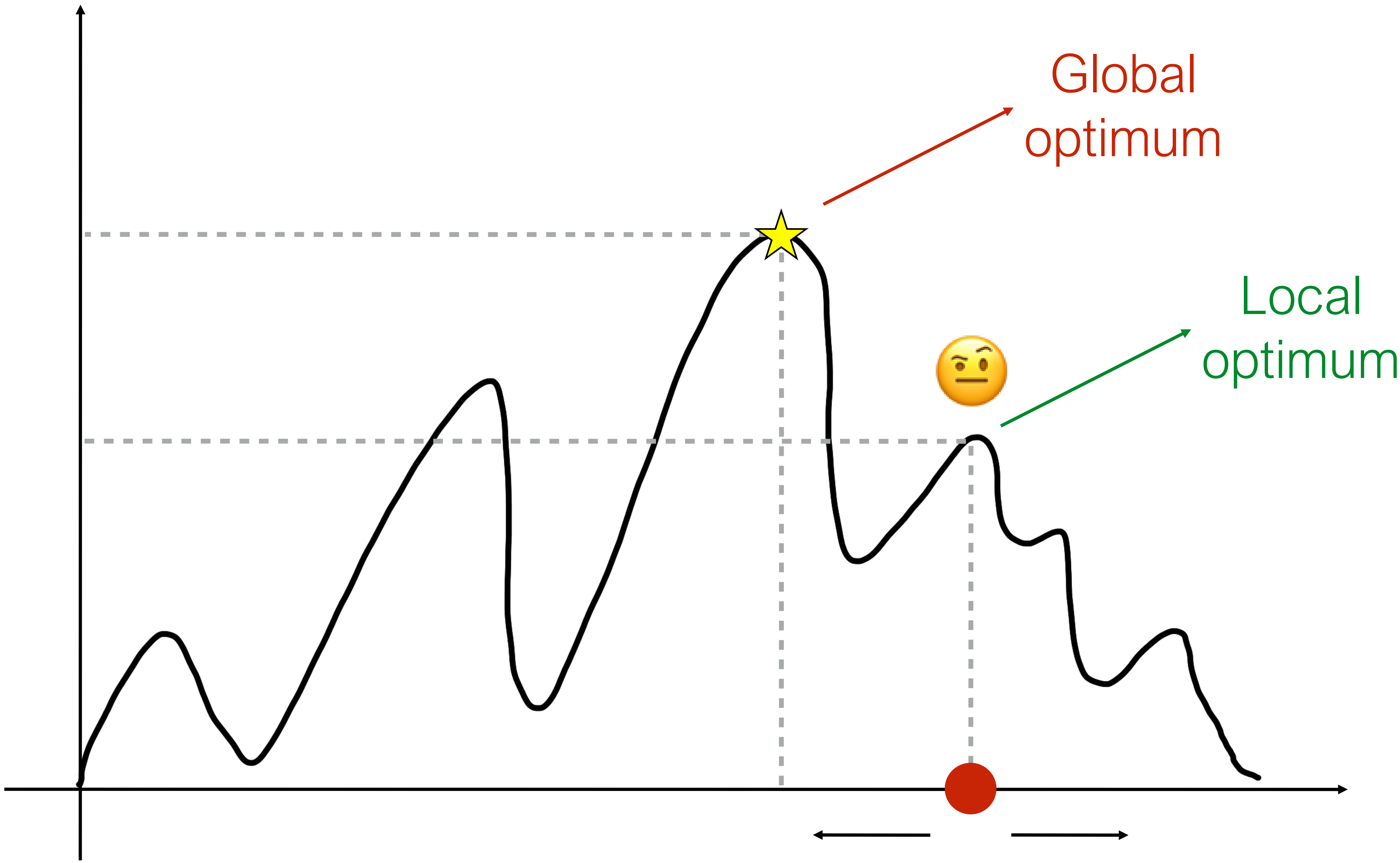
2
1
0
−1
−2
−1.0
−0.5
0.0
0.5
1.0

```python
def simple_hill_climbing(function, x_0, iterations):
    x = x_0
    u = 0.001
    for i in range(iterations):
        x_left, x_right = x-u, x+u
        y_left, y_right = function(x_left), function(x_right)
        if y_left > y_right:
            x = x_left
        elif y_right > y_left:
            x = x_right
        else:
            break
    return x
```



Stuck at local optima.

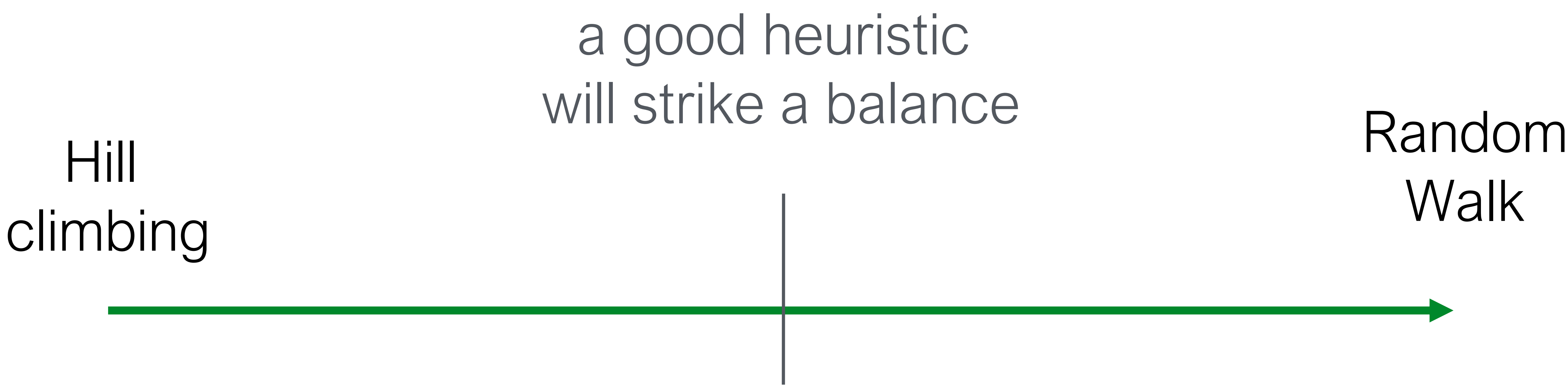
Hill climbing finds optimal solutions for <u>convex problems</u> – for other problems it will find only <u>local optima</u>



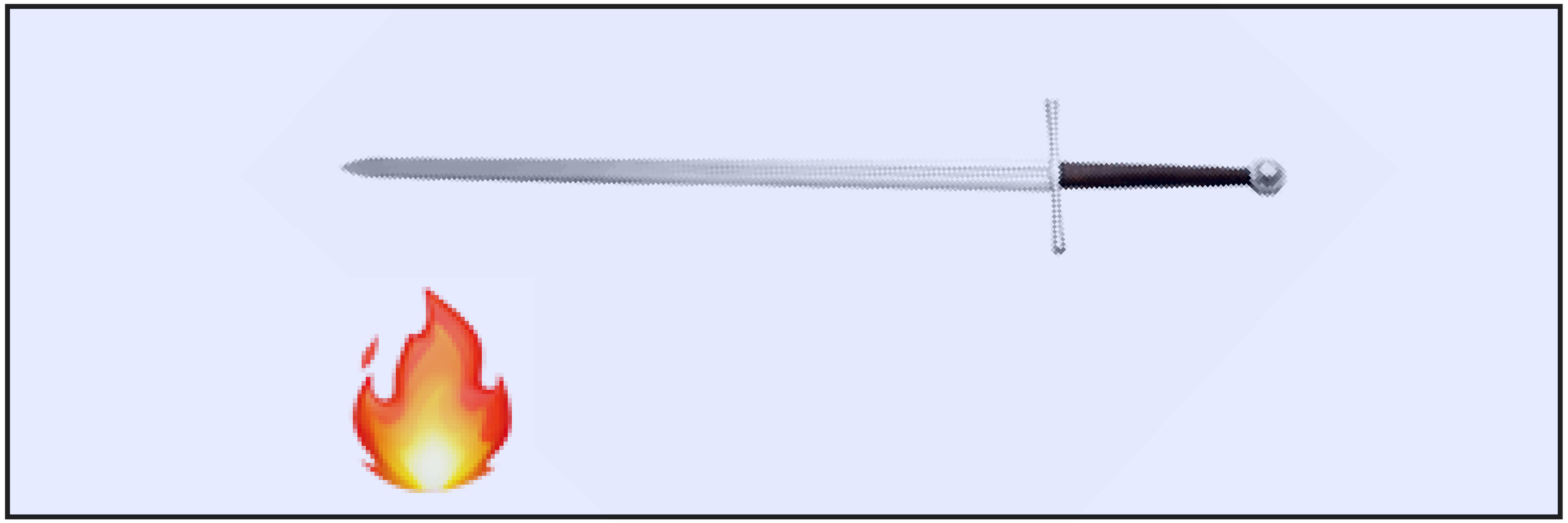
How to avoid local optima?

exploitation:
probing a limited (but promising) region of the search space in order to improve a promising solution.

exploration: probing a much larger portion of the search space with the hope of finding other promising solutions that are yet to be refined.



Annealing: repeated healing and cooling.



apply heat

let it cool

apply heat

let it cool

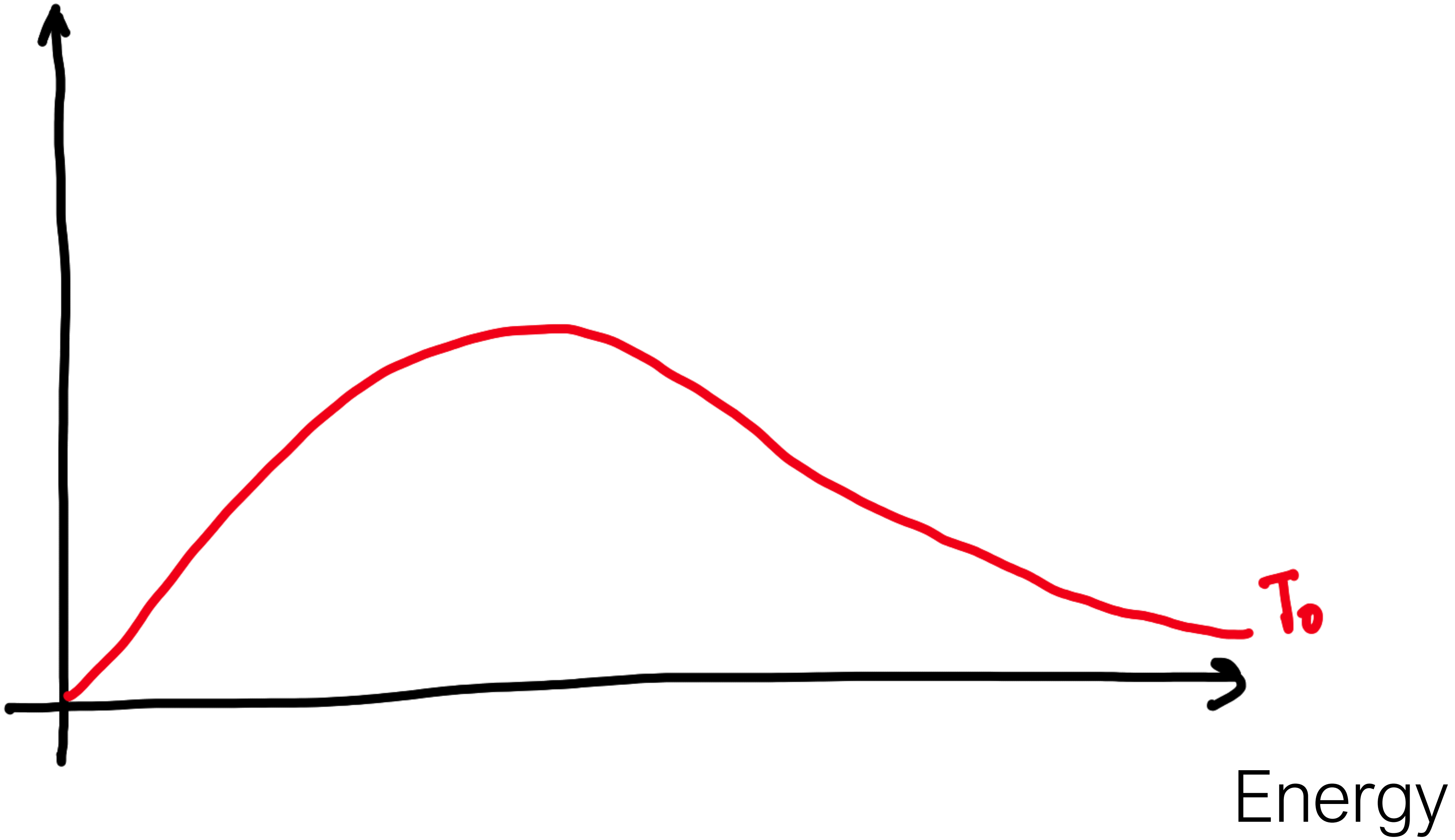
apply heat

How to simulate thermal equilibrium in a solid?

Metropolis, Rosenbluth, Rosenbluth, Teller, Teller. "Equation of state calculation by fast computing machines", Journal of Chemical Physics 21, 1087-1092, 1953

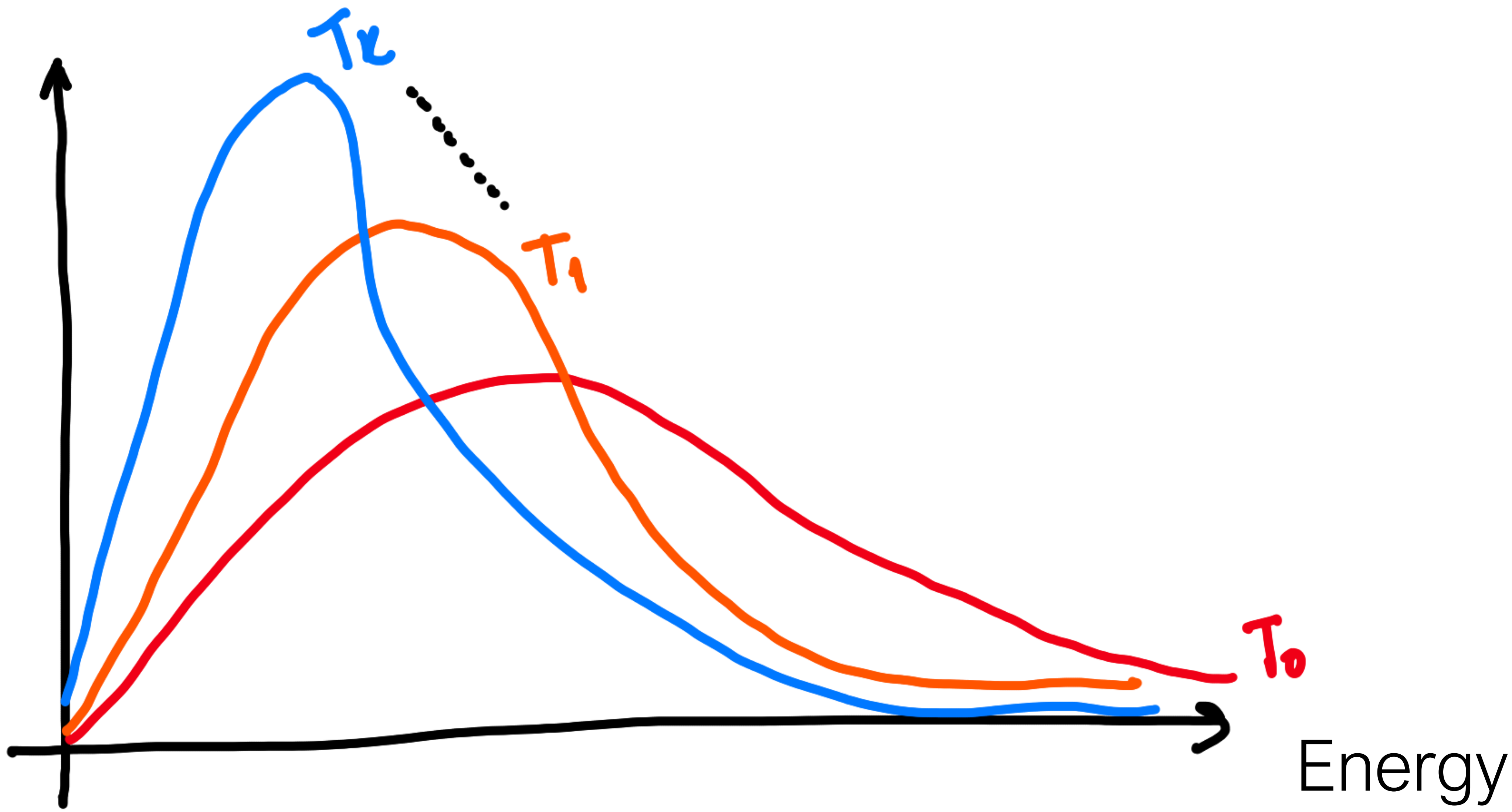
For a fixed temperature, the evolution towards "thermal equilibrium" is simulated using the Metropolis algorithm. This involves taking many samples (a.k.a Montecarlo), which are *accepted* or *rejected* based on the Boltzmann distribution. **Metropolis criterion.**

Boltzmann Distribution



$$P(E = E_a) \propto \exp\left(-\frac{E_a}{RT_0}\right)$$

$$T_k < T_{k-1} < \cdots < T_0$$



$$P(E = E_a) \propto \exp\left(-\frac{E_a}{RT}\right)$$

Simulated annealing — Idea

- At high temperatures, you are more likely to explore less *fit solutions*
- At low temperatures, you are less likely to explore less *fit solutions*
- Decrease the temperature as you perform the exploration of the search space

Hill climbing

Random Walk

Temperature

Simulated annealing — Algorithm

For a fixed temperature T:

1. Let x_i be the current solution to the problem.
2. Generate a perturbed solution $\tilde{x}$
3. Decide if $\tilde{x}$ is accepted or rejected.
4. If accepted, update new solution $x_{i+1} = \tilde{x}$, otherwise $x_{i+1} = x_i$
5. Repeat the process for many perturbations.

Decrease T.

For a fixed temperature T:

1. Let x_i be the current solution to the problem.
2. Generate a perturbed solution $\tilde{x}$
3. Decide if $\tilde{x}$ is accepted or rejected.
4. If accepted, update new solution $x_{i+1} = \tilde{x}$, otherwise $x_{i+1} = x_i$
5. Repeat the process for many perturbations.

Decrease T.

Metropolis criterion for maximising $f(x)$

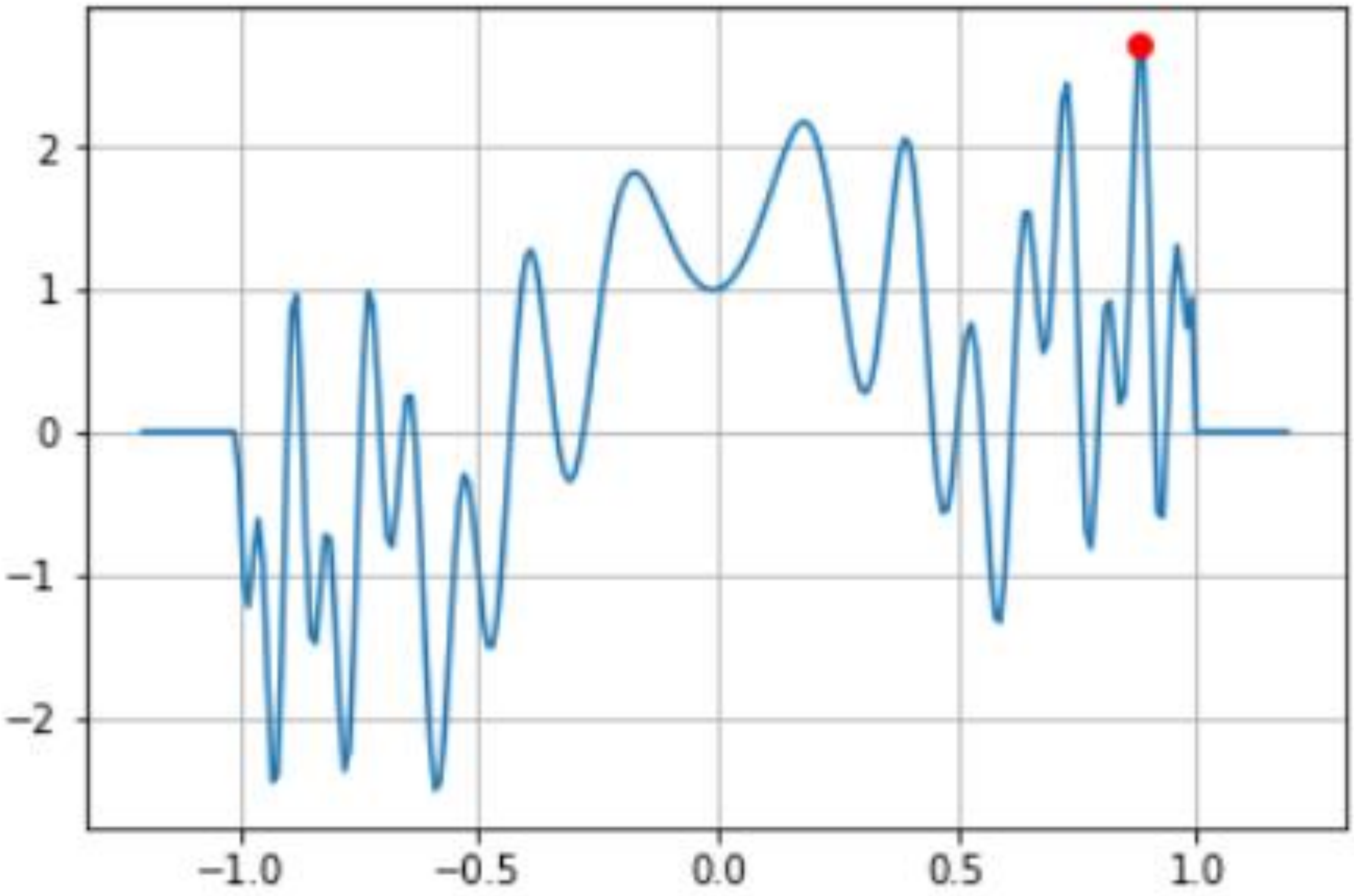
Accept with probability

$$\begin{cases} 1 & \text{if} \quad f(\tilde{x}) > f(x_i) \\ \exp\left(\frac{f(\tilde{x}) - f(x_i)}{T}\right) & \text{otherwise} \end{cases}$$

```python
def SA(function, search_space, perturbations_per_annealing_sep, t0, cooling_factor):
    assert t0 > 0
    assert 0 < cooling_factor < 1
    current_solution = np.random.choice(search_space)
    t = t0
    while t > 0.001:
        for _ in range(perturbations_per_annealing_sep):
            current_value = function(current_solution)

            perturbation = np.random.choice(search_space)
            perturbation_value = function(perturbation)

            delta = perturbation_value - current_value
            if delta > 0:
                current_solution = perturbation
                current_value = perturbation_value
            elif np.random.rand() < np.exp(delta/t):
                current_solution = perturbation
                current_value = perturbation_value
        t = cooling_factor*t
    return current_solution, function(current_solution)
```



2
1
0
−1
−2
−1.0
−0.5
0.0
0.5
1.0

Metropolis criterion for maximising $f(x)$

Accept with probability
$$\begin{cases} 1 & \text{if} \quad f(\tilde{x}) > f(x_i) \\ \exp\left(\dfrac{f(\tilde{x}) - f(x_i)}{T}\right) & \text{otherwise} \end{cases}$$

- If the perturbed value is an improvement, take it.
- If the values are infinitesimally close, take the perturbation.
- If it’s a big step down, don’t take the perturbation.
- What is the role of T:
    - Large T — very likely to accept
    - T Close to 0 — don’t take downward steps

Practical considerations

- Cooling schedule: How to decrease T. Generally *ad hoc.*
- No systematic way — generally true of heuristics.
- Common pattern: $T_{i+1} = \alpha T_i \quad 0 < \alpha < 1$
- Go slow $0.8 < \alpha < 0.99$, start with large T
- As many perturbations as practical.
- Faster cooling, more repetitions…

https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.optimize.anneal.html

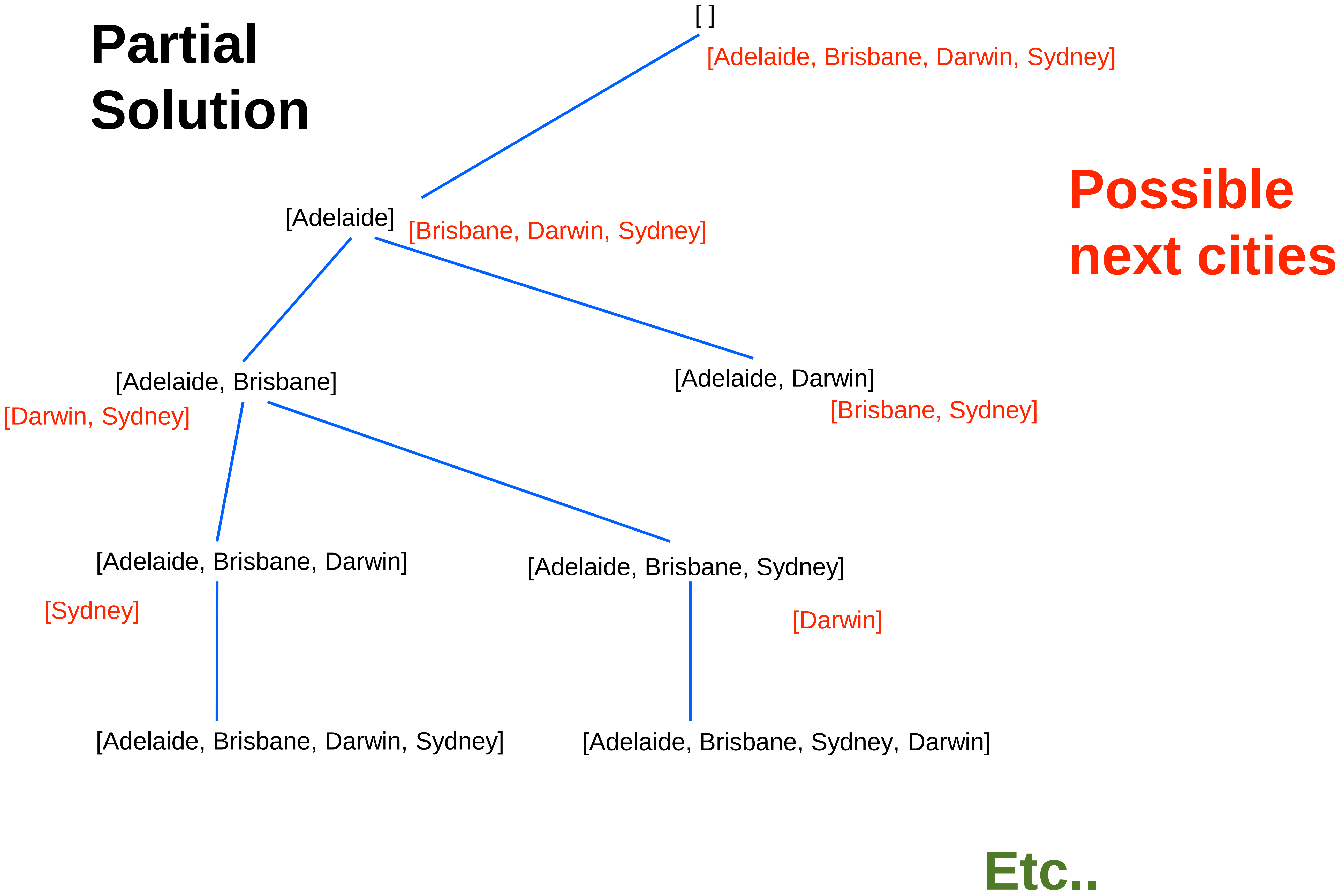
Traveling Salesman

Suppose you are given the following driving distances in kms between the following capital cities.

distance

	Adelaide	Brisbane	Canberra	Darwin	Sydney
Adelaide		2053	1155	3017	1385
Brisbane	2053		1080	3415	939
Canberra	1155	1080		3940	285
Darwin	3017	3415	3940		3975
Sydney	1385	939	285	3975	

Find the shortest route that enables a salesman to <u>start at Canberra</u>, visit all the other cities, before <u>returning to Canberra.</u>


Partial
Solution
[ ]
[Adelaide, Brisbane, Darwin, Sydney]
Possible
next cities
[Adelaide]
[Brisbane, Darwin, Sydney]
[Adelaide, Brisbane]
[Darwin, Sydney]
[Adelaide, Darwin]
[Brisbane, Sydney]
[Adelaide, Brisbane, Darwin]
[Adelaide, Brisbane, Sydney]
[Sydney]
[Darwin]
[Adelaide, Brisbane, Darwin, Sydney]
[Adelaide, Brisbane, Sydney, Darwin]
Etc..

Recommended Reading

Chapter 2, P.J.M. van Laarhoven and E.H.L. Aarts, **Simulated Annealing: Theory and Applications**. D. Reidel Publishing Company.

Please share pictures of your pets on the EDForum Post!