Optimization & Local Search

Alexander G. Ororbia II

Biologically-Inspired Intelligent Systems

CSCI-633

1/18/2024

So… what is mathematical optimization, anyway?

"Optimization" comes from the same root as "optimal", which means *best*. When you optimize something, you are "making it best".

But "best" can vary. If you're a football player, you might want to maximize your running yards, and also minimize your fumbles. Both maximizing and minimizing are types of optimization problems.

In the modern world, pennies matter, microseconds matter, microns matter.

Categories of Decision making problems

Category 1:

- The set of possible alternatives for the decision is a finite discrete set typically consisting of a small number of elements.
  - Example: “A teenage girl knows four boys all of whom she likes, and has to decide who among them to go steady with.”
- Solution: scoring methods

Category 2:

- The number of possible alternatives is either infinite, or finite but very large, and the decision may be required to satisfy some restrictions and constraints
- Solution: unconstrained and constrained optimization methods

Category 2 Decision Problems & Solution Flow

1. Get a precise definition of the problem, all relevant data and information on it.
    - Uncontrollable factors (random variables)
    - Controllable inputs (decision variables)

2. Construct a mathematical (optimization) model of the problem.
    - Build objective functions and constraints

3. Solve the model
    - Apply the most appropriate algorithms for the given problem

4. Implement the solution

Mathematical Optimization in the "Real World"

Mathematical Optimization is a branch of applied mathematics which is useful in many different fields. Here are a few examples:

- Manufacturing
- Production
- Inventory control
- Transportation
- Scheduling
- Networks
- Finance
- Engineering
- Mechanics
- Economics
- Control engineering
- Marketing
- Policy Modeling

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.
- Variables, x_1 x_2 x_3 and so on, which are the inputs – things you can control. They are abbreviated x_n to refer to individuals or x to refer to them as a group.

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.
- Variables, x_1 x_2 x_3 and so on, which are the inputs – things you can control. They are abbreviated x_n to refer to individuals or x to refer to them as a group.
- Constraints, which are equations that place limits on how big or small some variables can get. Equality constraints are usually noted $h_n(x)$ and inequality constraints are noted $g_n(x)$.

Optimization Vocabulary

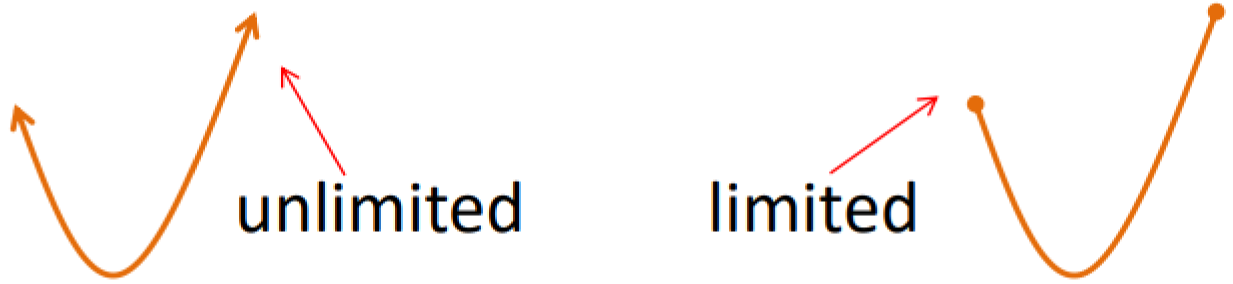
A football coach is planning practices for his running backs.

- His main goal is to maximize running yards – this will become his objective function.
- He can make his athletes spend practice time in the weight room; running sprints; or practicing ball protection. The amount of time spent on each is a variable.
- However, there are limits to the total amount of time he has. Also, if he completely sacrifices ball protection he may see running yards go up, but also fumbles, so he may place an upper limit on the amount of fumbles he considers acceptable. These are constraints.

Note that the variables influence the objective function and the constraints place limits on the domain of the variables.

Types of Optimization Problems

- Some problems have constraints and some do not.



Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.

x_3

x_1

x_4 x_2

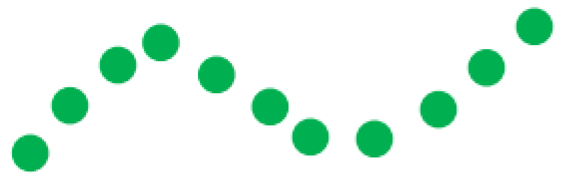
x_6 x_5

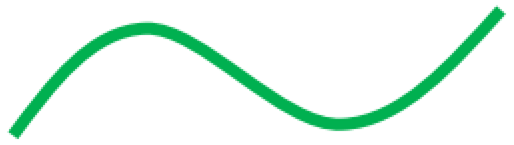
x_7

x_8

Types of Optimization Problems

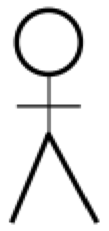
- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.

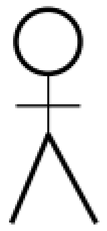




Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).



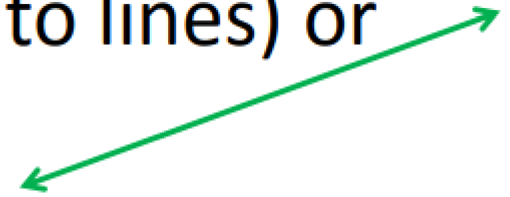


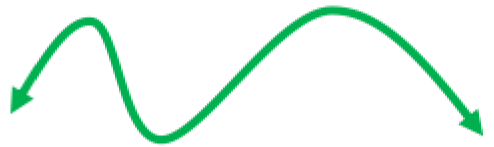
Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).
- Systems can be deterministic (specific causes produce specific effects) or stochastic (involve randomness/ probability).

Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).
- Systems can be deterministic (specific causes produce specific effects) or stochastic (involve randomness/ probability).
- Equations can be linear (graph to lines) or nonlinear (graph to curves)





Convex vs. non-convex optimization problems!

Why Mathematical Optimization is worth learning

Q: Which of these things is not like the others?

a) A degree in engineering

b) A degree in chemistry

c) A degree in pure mathematics

d) A large pepperoni pizza

Why Mathematical Optimization is worth learning

Q: Which of these things is not like the others?

a) A degree in engineering

b) A degree in chemistry

[✓ c) A degree in pure mathematics]

d) A large pepperoni pizza

(With the others, you can feed a family of four)

Optimization at a Glance

- Optimization is the act of obtaining the best result under given circumstances.

- Optimization can be defined as the process of finding the conditions that give the maximum or minimum of a function.

- The optimum seeking methods are also known as *mathematical programming techniques* and are generally studied as a part of operations research.

- *Operations research* is a branch of mathematics concerned with the application of scientific methods and techniques to decision making problems and with establishing the best or optimal solutions.

Problem Specification/Formulation

- **General mathematical optimization (minimization) problem:**

$$\text{minimize } f_i(\boldsymbol{x}), i = 1, 2, \ldots, M$$

$$\text{subject to } h_j(\boldsymbol{x}) = 0, \quad j = 1, 2, \ldots, J$$
$$g_k(\boldsymbol{x}) \leq 0, \quad k = 1, 2, \ldots, K$$

- $f_i : \boldsymbol{R^d} \rightarrow \boldsymbol{R}$**:** objective/cost fnctn (maps search/design space -> solution/response space)
- $\boldsymbol{x}=(x_1, \ldots\ldots, x_d)^T$: design variables - unknowns of the problem, could be mix of discrete & continuous (contus) values (***x*** is "design vector")
- $h_j : \boldsymbol{R^d} \rightarrow \boldsymbol{R}$**:** equality constraints
- $g_k : \boldsymbol{R^d} \rightarrow \boldsymbol{R}$**:** *in*equality constraints

- This problem is a constrained optimization problem
  - Linear constraints + linear objectives → linear programming problem

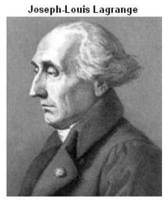
Some History

Historical development

- Isaac Newton (1642-1727)
  (The development of differential calculus
  methods of optimization)

- Joseph-Louis Lagrange (1736-1813)
  (Calculus of variations, minimization of functionals,
  method of optimization for constrained problems)

- Augustin-Louis Cauchy (1789-1857)
  (Solution by direct substitution, steepest
  descent method for unconstrained optimization)



Some History

Historical development

- Leonhard Euler (1707-1783)
  (Calculus of variations, minimization of
  functionals)

- Gottfried Leibnitz (1646-1716)
  (Differential calculus methods
  of optimization)

İsim: Gottfired Wilhelm von Leibniz

Some History

Historical development

- George Bernard Dantzig (1914-2005)
  (Linear programming and Simplex method (1947))

- Richard Bellman (1920-1984)
  (Principle of optimality in dynamic
  programming problems)

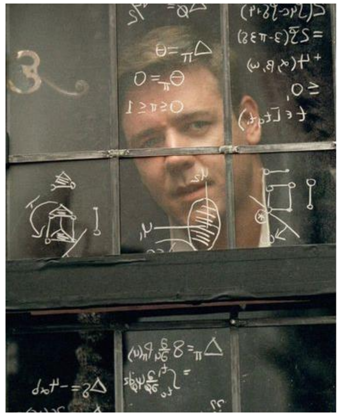
- Harold William Kuhn (1925-)
  (Necessary and sufficient conditions for the optimal solution of programming problems, game theory)

Some History

Historical development

- Albert William Tucker (1905-1995)
  (Necessary and sufficient conditions
  for the optimal solution of programming
  problems, nonlinear programming, game
  theory: his PhD student
  was John Nash)

- Von Neumann (1903-1957)
  (game theory)



Equivalence between Minimum and Maximum

- If a point *x** corresponds to the minimum value of the function *f* (*x*), the same point also corresponds to the maximum value of the negative of the function, -*f* (*x*).
  - This means optimization can be re-interpreted to mean minimization since the maximum of a function can be found by seeking the minimum of the negative of the same function.

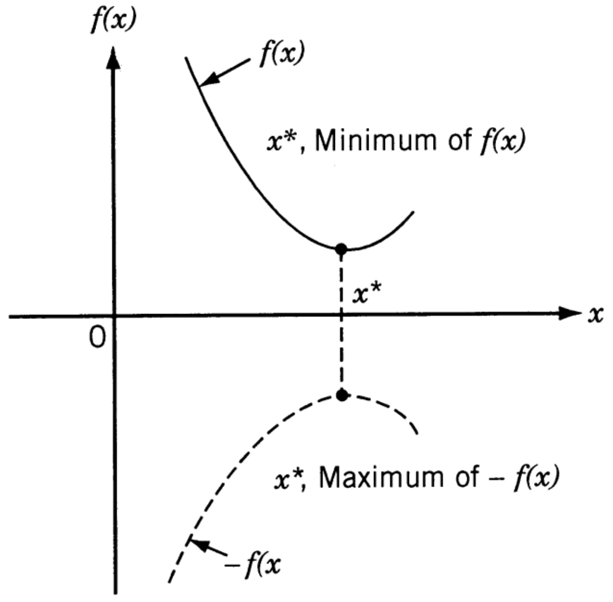
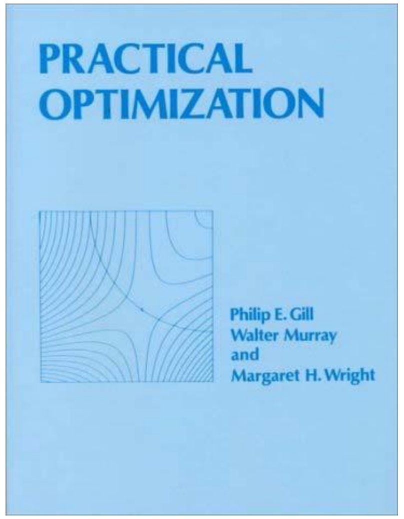


Figure 1.1 Minimum of $f(x)$ is same as maximum of $-f(x)$.

Books to Read

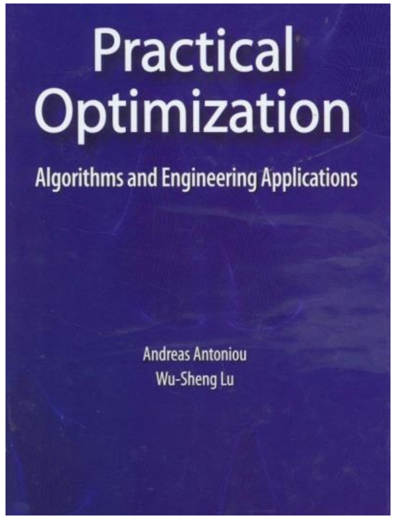
- **Practical Optimization**
  - Philip E. Gill, Walter Murray, and Margaret H. Wright, Academic Press, 1981



PRACTICAL OPTIMIZATION

Philip E. Gill
Walter Murray
and
Margaret H. Wright

- **Practical Optimization: Algorithms and Engineering Applications**
  - Andreas Antoniou and Wu-Sheng Lu, 2007



Practical Optimization
Algorithms and Engineering Applications

Andreas Antoniou
Wu-Sheng Lu

- Both cover unconstrained and constrained optimization. Very clear and comprehensive.

Optimization: A Local Search Perspective

Local Search Algorithms

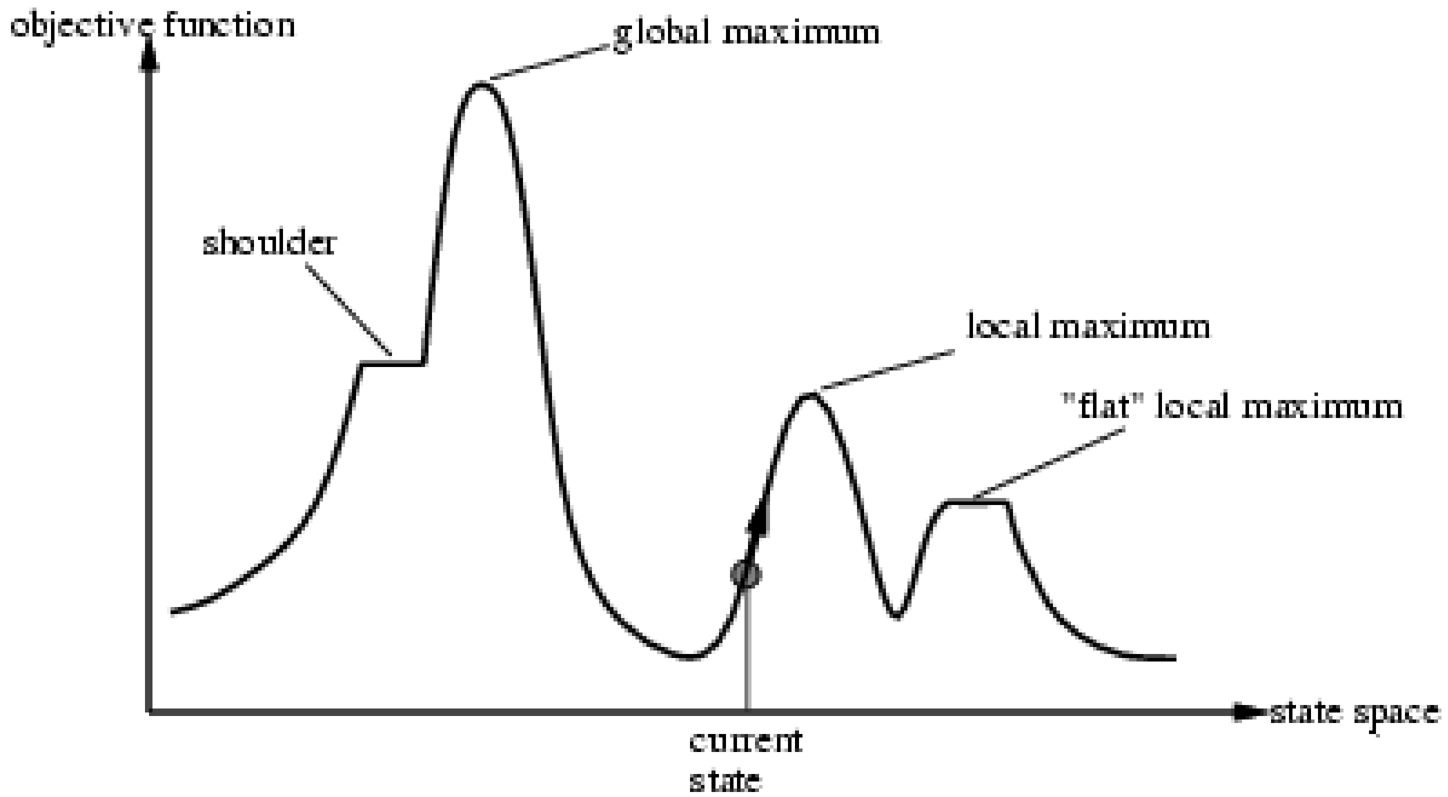
- Optimization Problems
  - Path to goal often irrelevant; goal state is the solution (e.g. n-queens problem, training neural networks)
    - *Also good for problems w/ no goal test/path cost*
- State space (search space)
  - Represented by set of **complete** state configurations
  - Can be discrete or continuous (contus)
- Search Goal
  - Find configuration satisfying constraints, e.g., n-queens
- Local Search Algorithms
  - Keep a single "current" state (= candidate solution), improves it if possible (iterative improvement*)*

Local Search: Iterative improvement

- Start with a complete valid state
- Gradually work to improve to better and better states
    - Sometimes, try to achieve an optimum, though not always possible
- Sometimes states are discrete, sometimes continuous

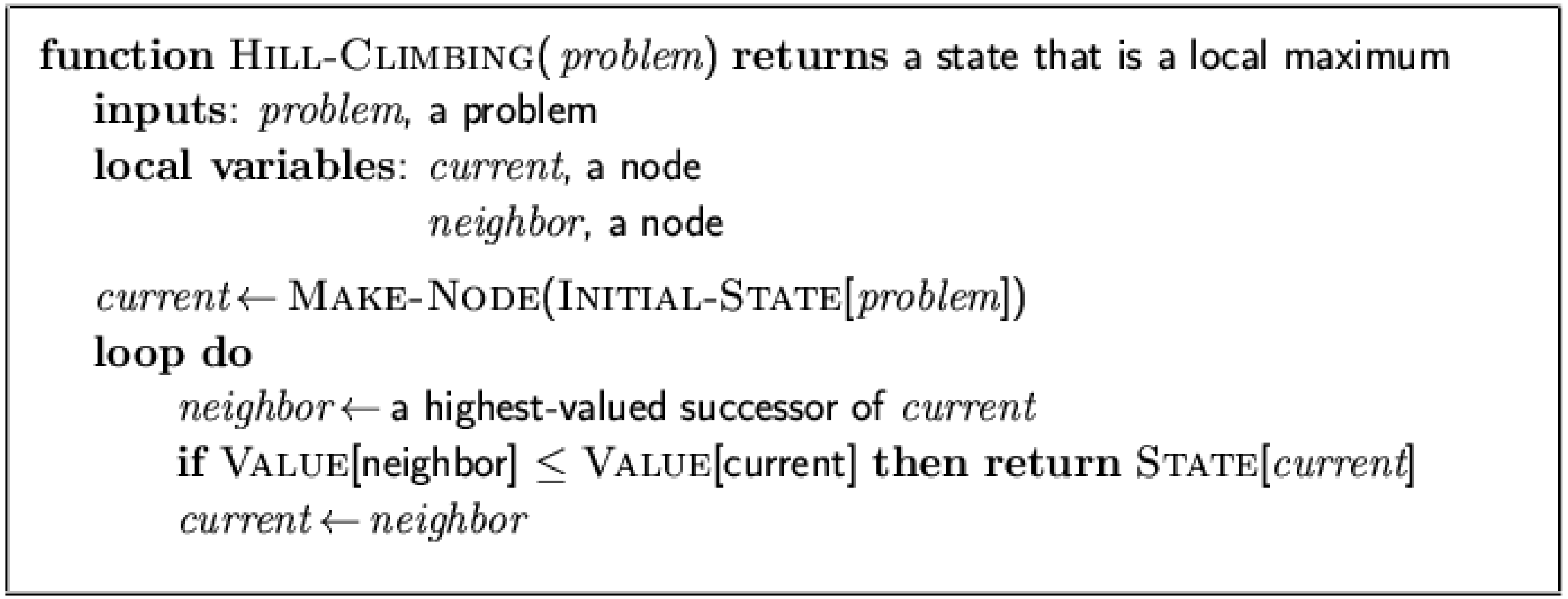
Optimization & State-Space Landscape

- **Goal types**:
    - Objective function -> find global max, find global min (usually not possible, so local)
- Complete = finds optima if it exists, Optimal = finds global optima



Hill-Climbing Search

- "Like climbing Everest in a thick fog with amnesia”, steepest ascent
- Terminates if cost/objective goes down



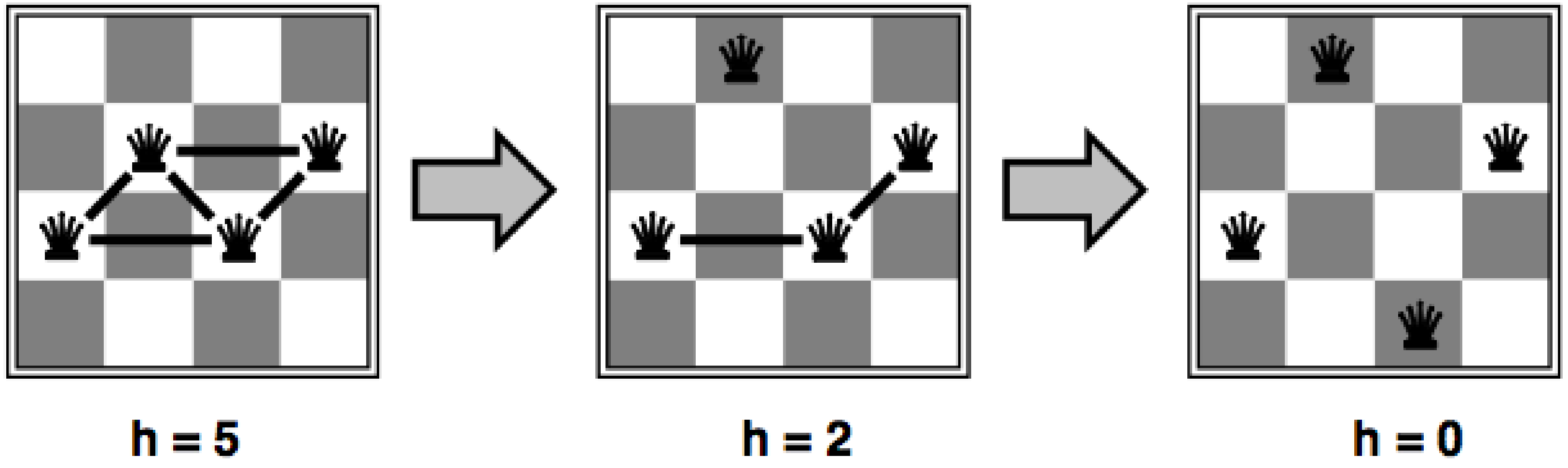
```
function HILL-CLIMBING(problem) returns a state that is a local maximum
    inputs: problem, a problem
    local variables: current, a node
                     neighbor, a node

    current ← MAKE-NODE(INITIAL-STATE[problem])
    loop do
        neighbor ← a highest-valued successor of current
        if VALUE[neighbor] ≤ VALUE[current] then return STATE[current]
        current ← neighbor
```

Example: n-queens

Put n queens on an $n \times n$ board with no two queens on the same row, column, or diagonal

Move a queen to reduce number of conflicts



Almost always solves n-queens problems almost instantaneously for very large n, e.g., $n = 1 million$

Variations of Hill-Climbing

- Stochastic hill climbing (selection probability depends on steepness of uphill move)
- First-choice hill-climbing = randomly generate successors until one is better than current
- All incomplete!
  - Unless use random restart (*if at first don't succeed, try, try again*), i.e., <u>*random restart hill-climbing*</u> = a # of restarts required proportional to probability of success *p* (or 1/p)

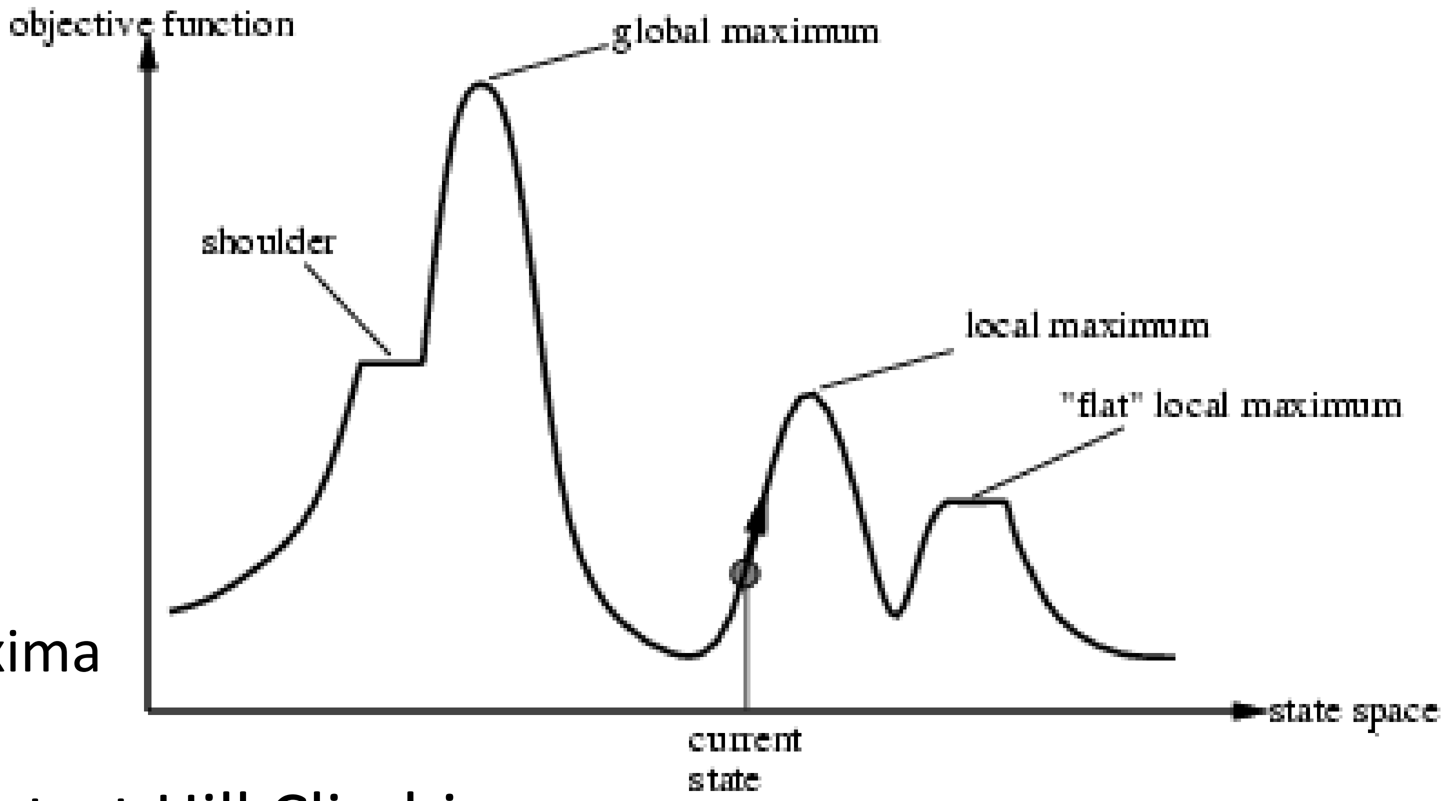
• Can work on *n*-million queens problem

- NP-hard problems have exponential number of local optima

• **The Hope**: find "good enough" local optima

Local Search Landscape & Climbing Hills

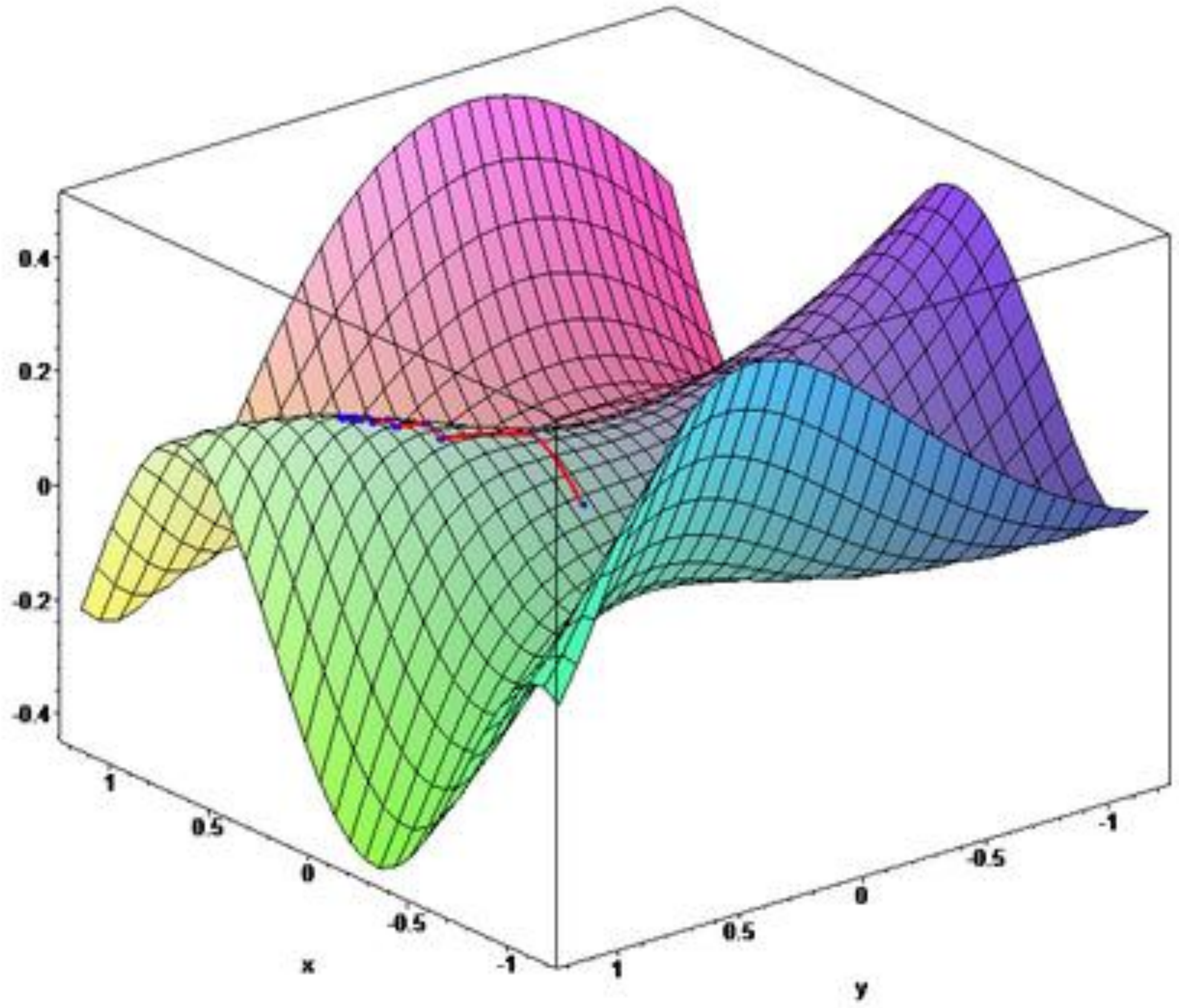
State Space Landscape



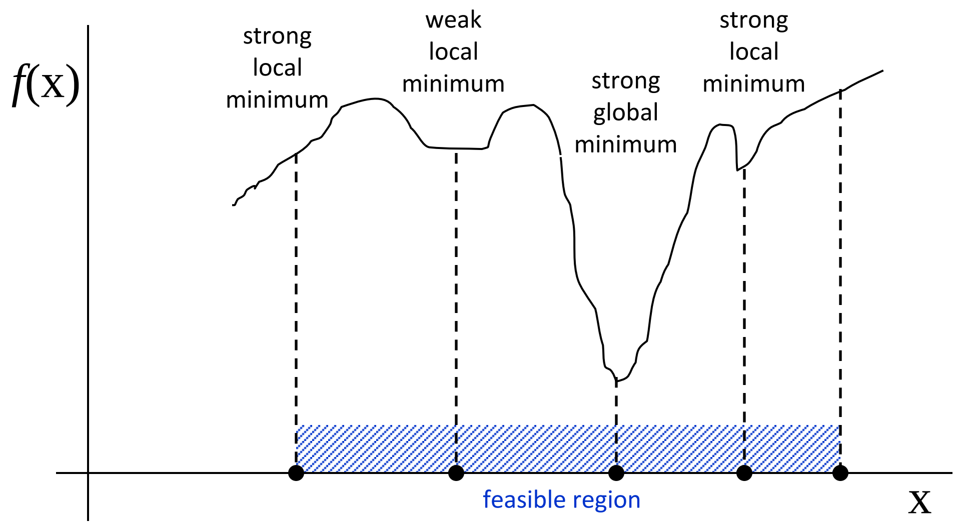
- •Problem
  - Stuck in
  - local maxima
- •Random-Restart Hill Climbing
  - Do series of hill-climbing searches from randomly chosen initial state

Why Optimization Again?

- Assume a state (or solution) with many variables
- Assume some function that you want to maximize/minimize value of
    - E.g. a “goodness” function
- Searching entire space is too complicated
    - Cannot evaluate every possible combination of variables
    - Function might be difficult to evaluate analytically

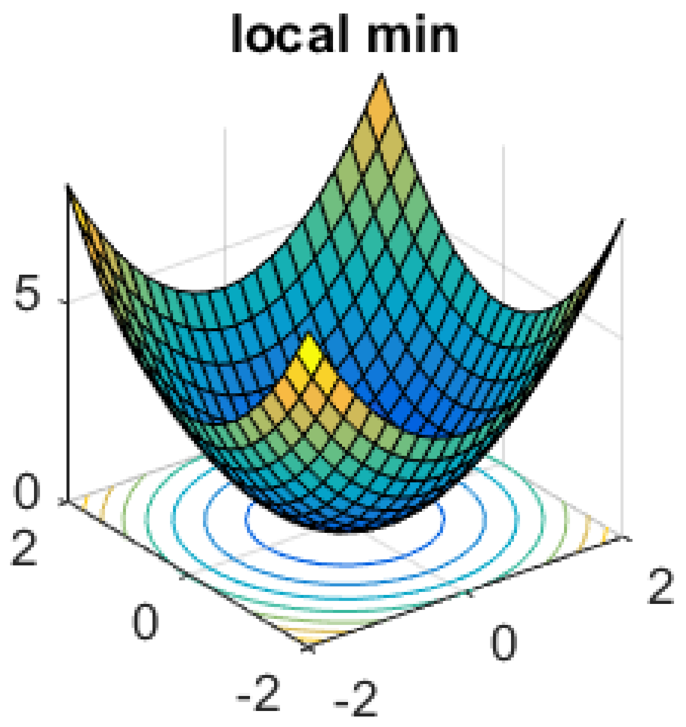

0.4
0.2
0
-0.2
-0.4
1
0.5
0
-0.5
-1
x
1
0.5
0
-0.5
-1
y

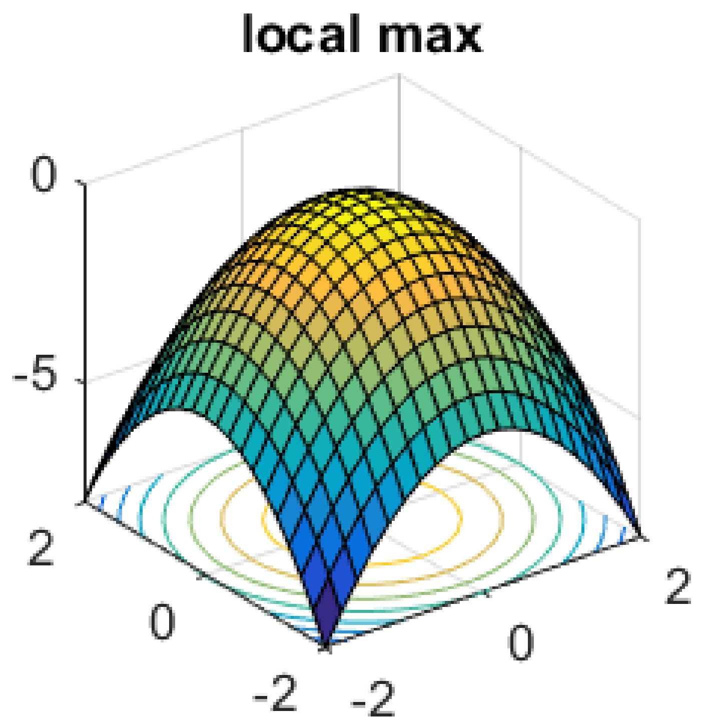
Types of Minima

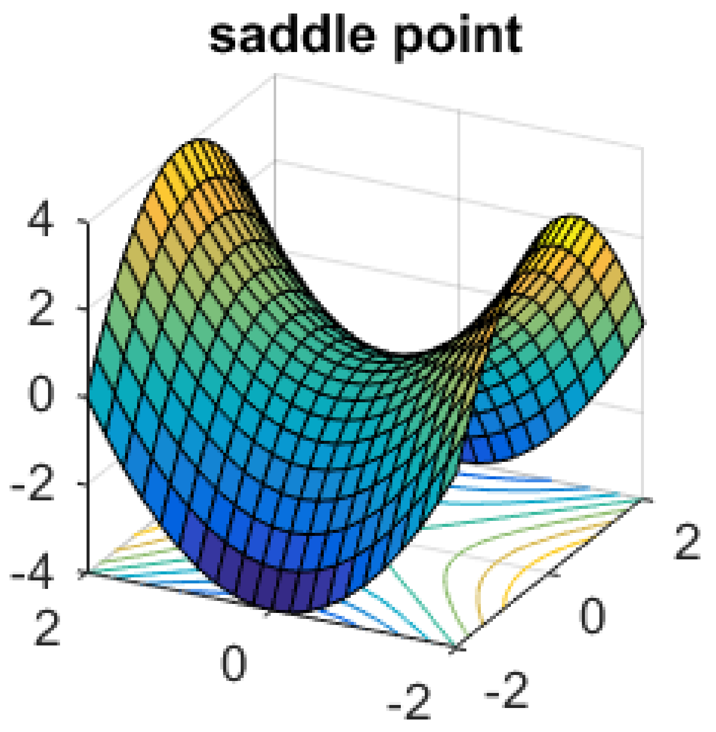


- Which of the minima is found depends on the starting point
- Such minima often occur in real applications

Problems!!

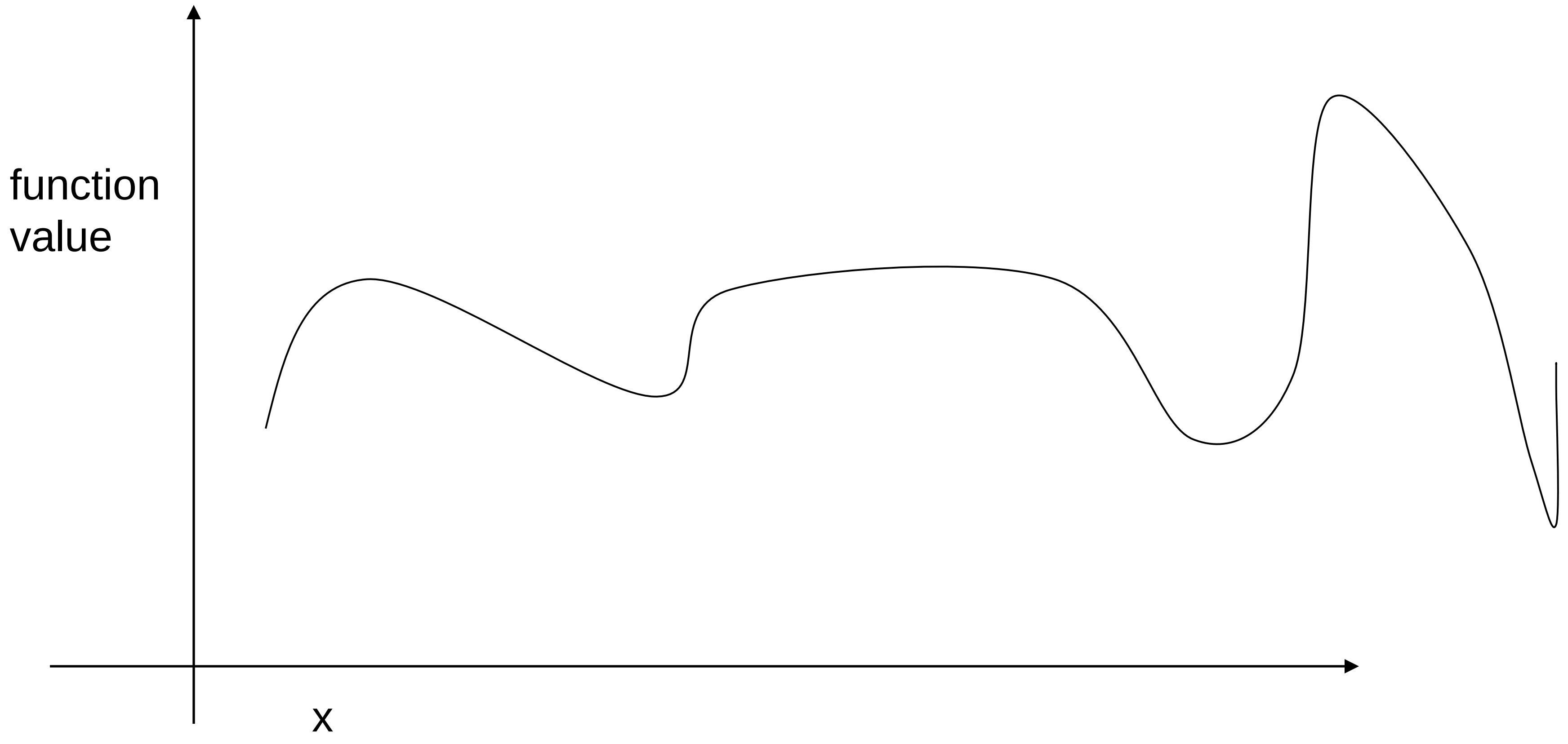

local min
5
0
2
0
-2
-2
0
2


local max
0
-5
2
0
-2
-2
0
2


saddle point
4
2
0
-2
-4
2
0
-2
-2
0
2

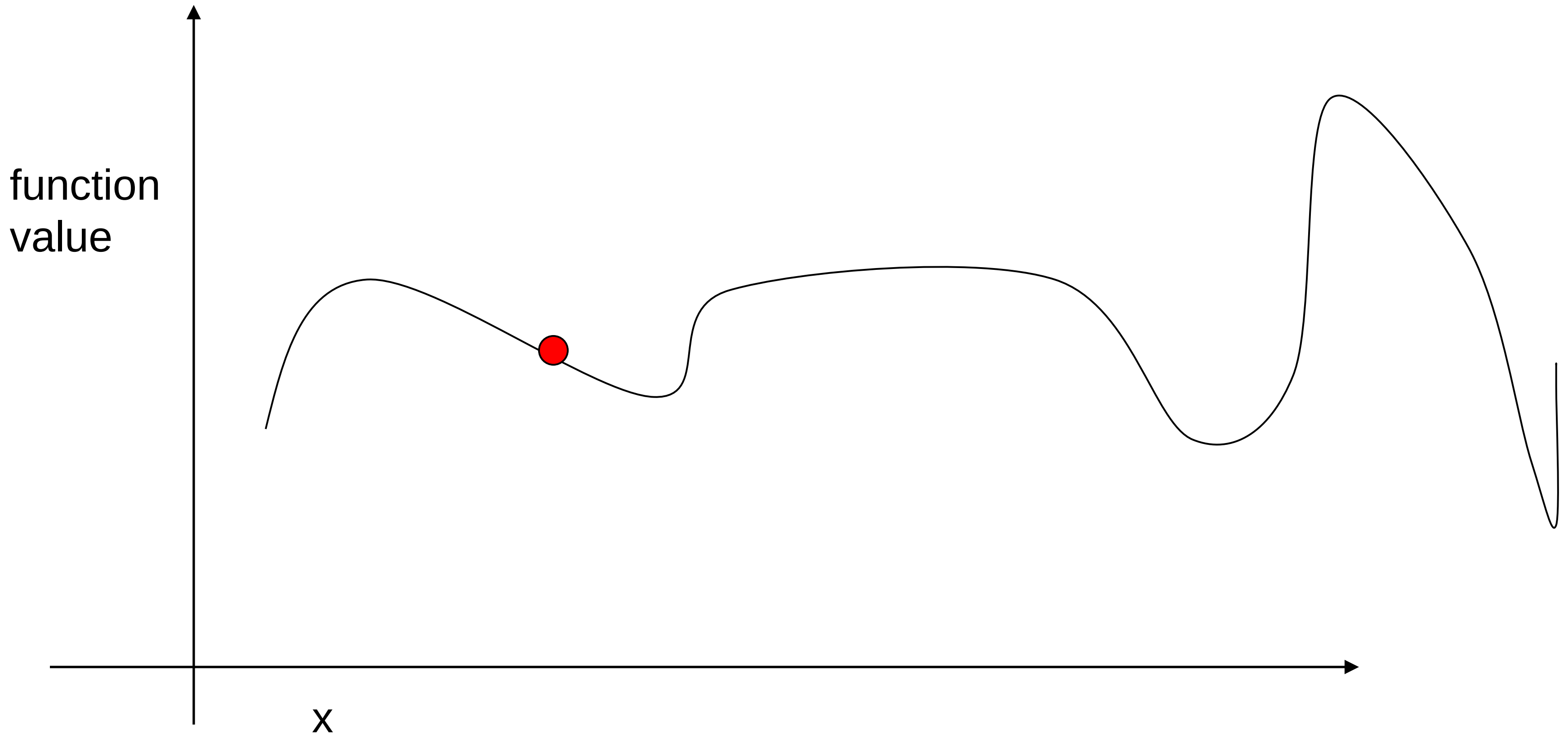
Simple Example: The Idealized Climb

- One dimension (typically use more):



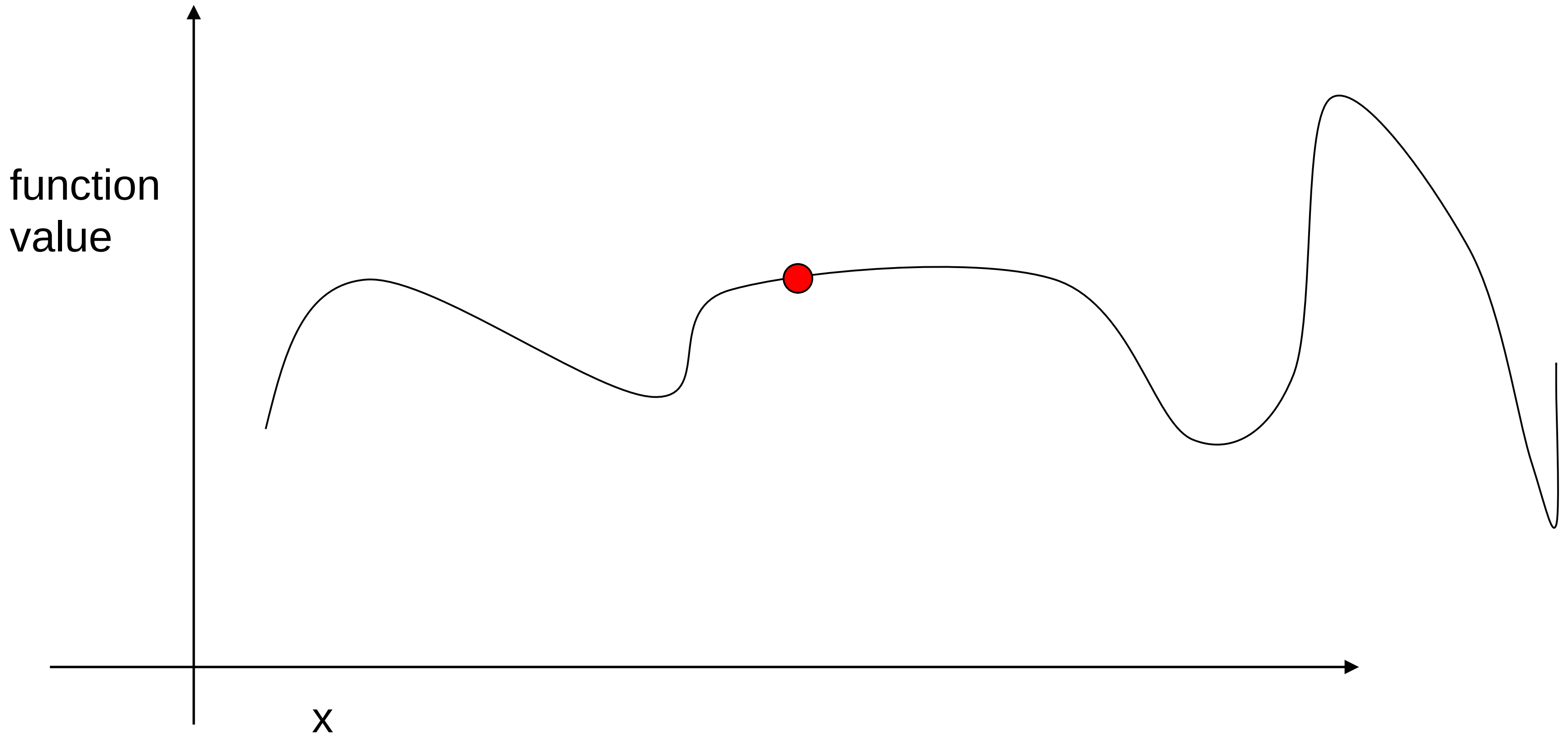
Simple Example: The Idealized Climb

- Start at a valid state, try to maximize



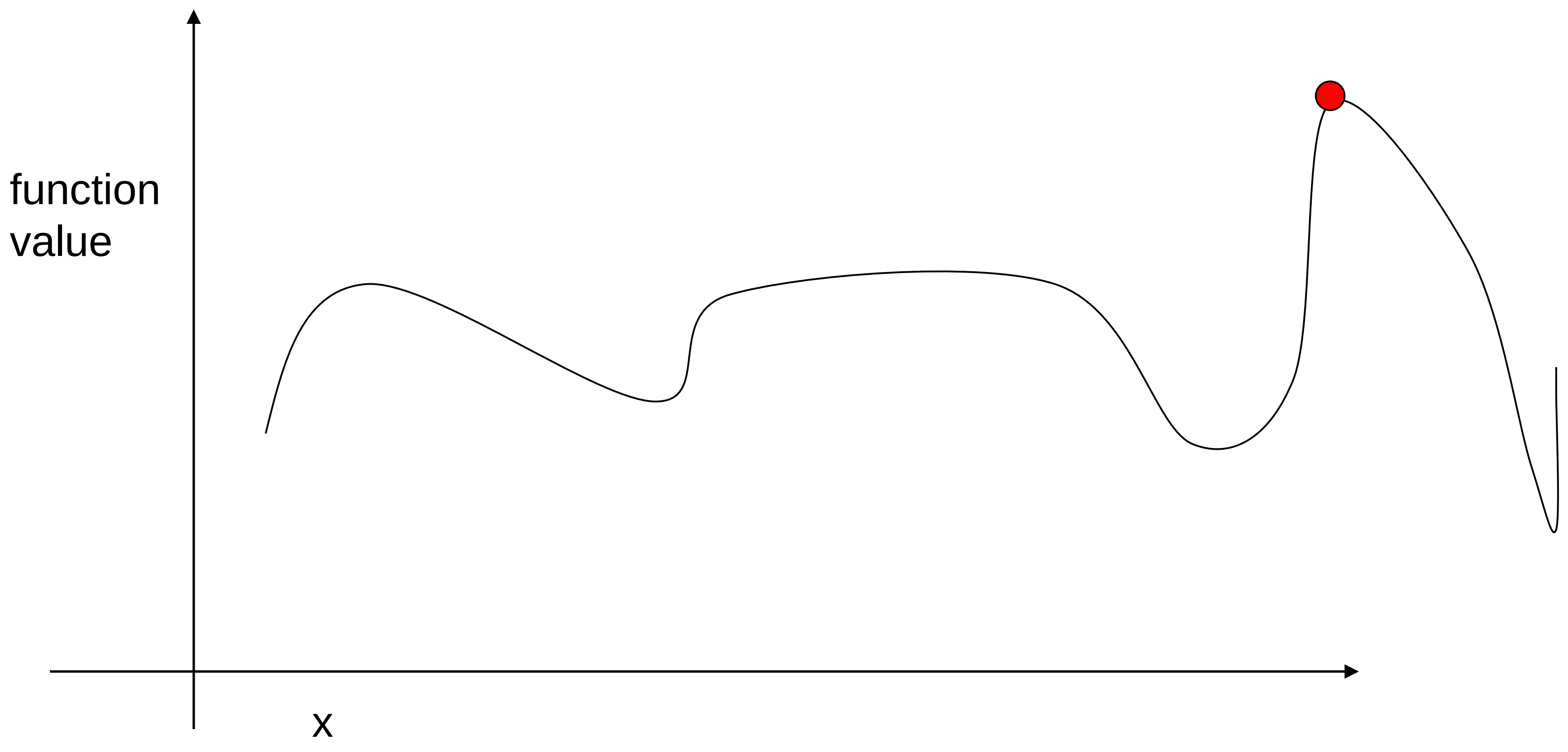
Simple Example: The Idealized Climb

- Move to better state



Simple Example: The Idealized Climb

- Try to find maximum



Questions?