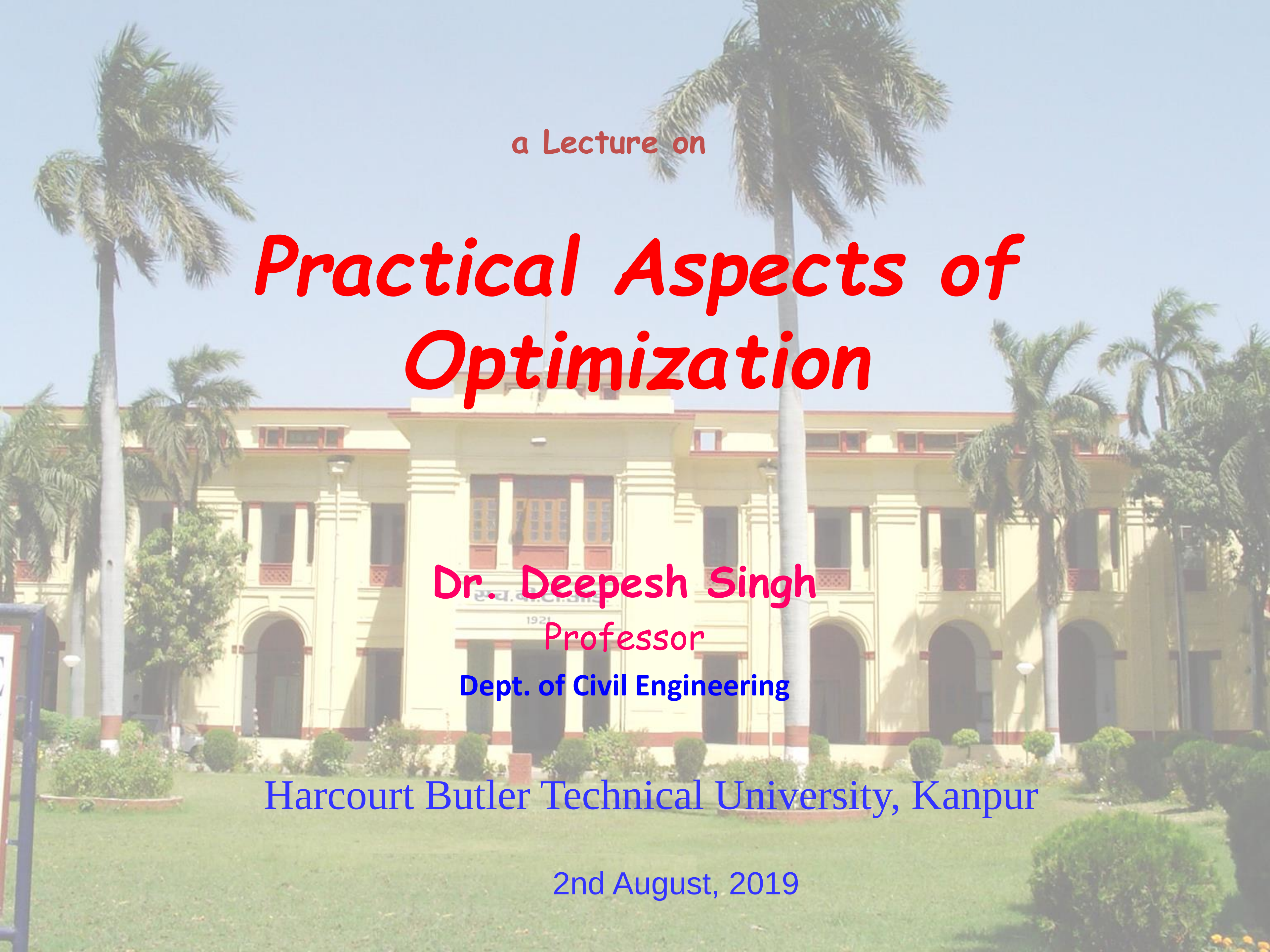


a Lecture on

# Practical Aspects of Optimization

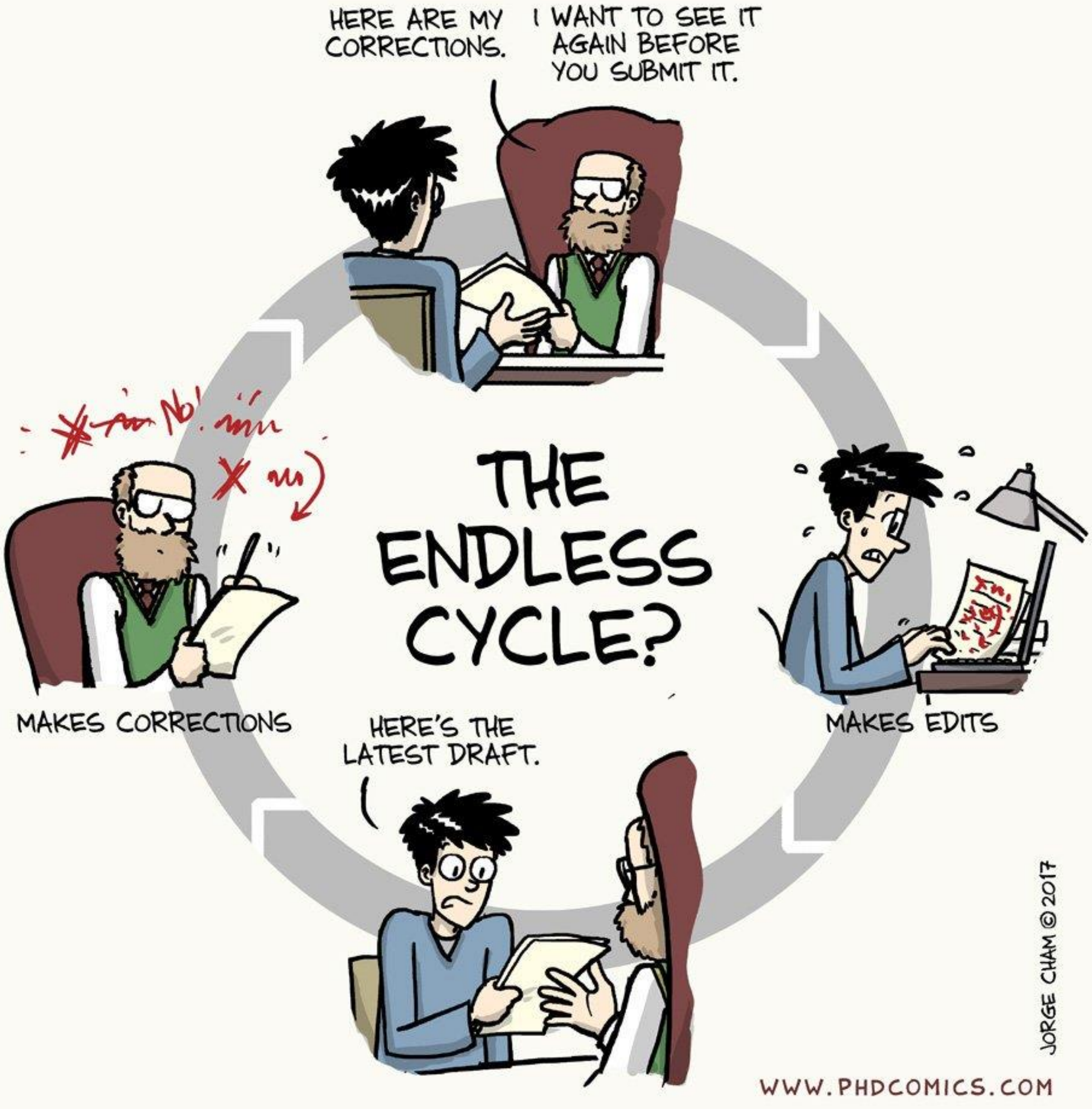
**Dr. Deepesh Singh**

Professor

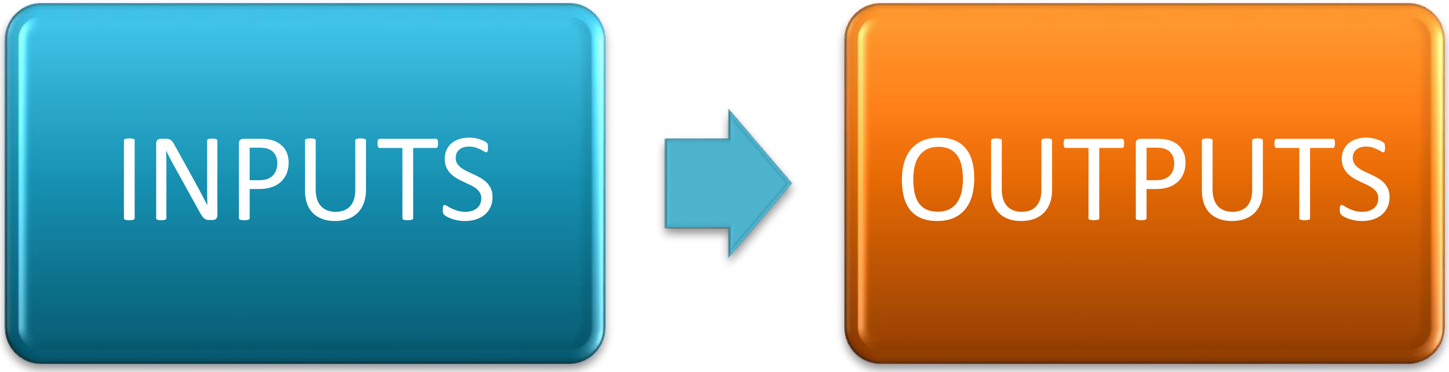
**Dept. of Civil Engineering**

Harcourt Butler Technical University, Kanpur

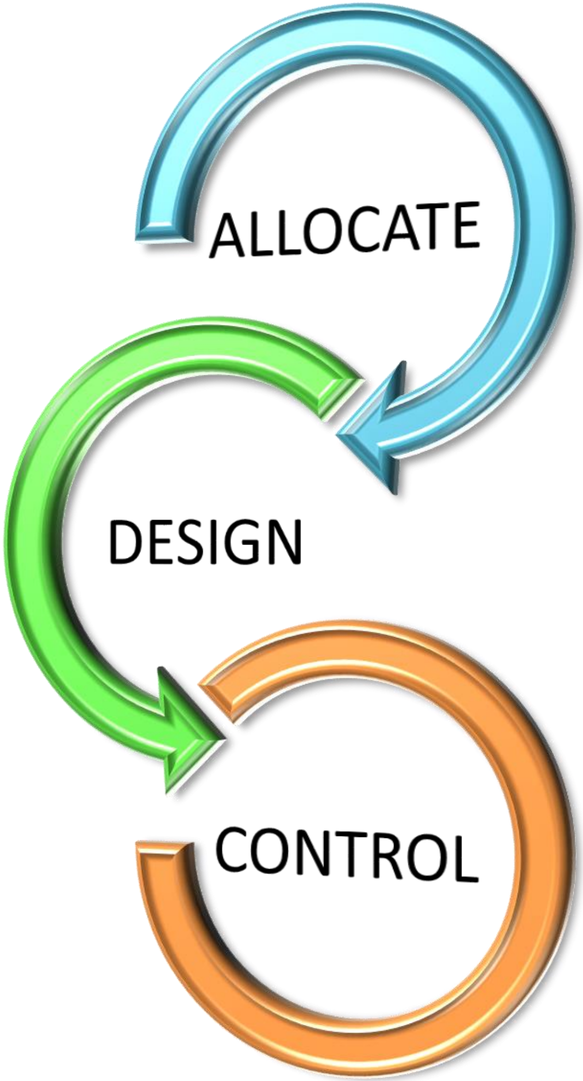
2nd August, 2019

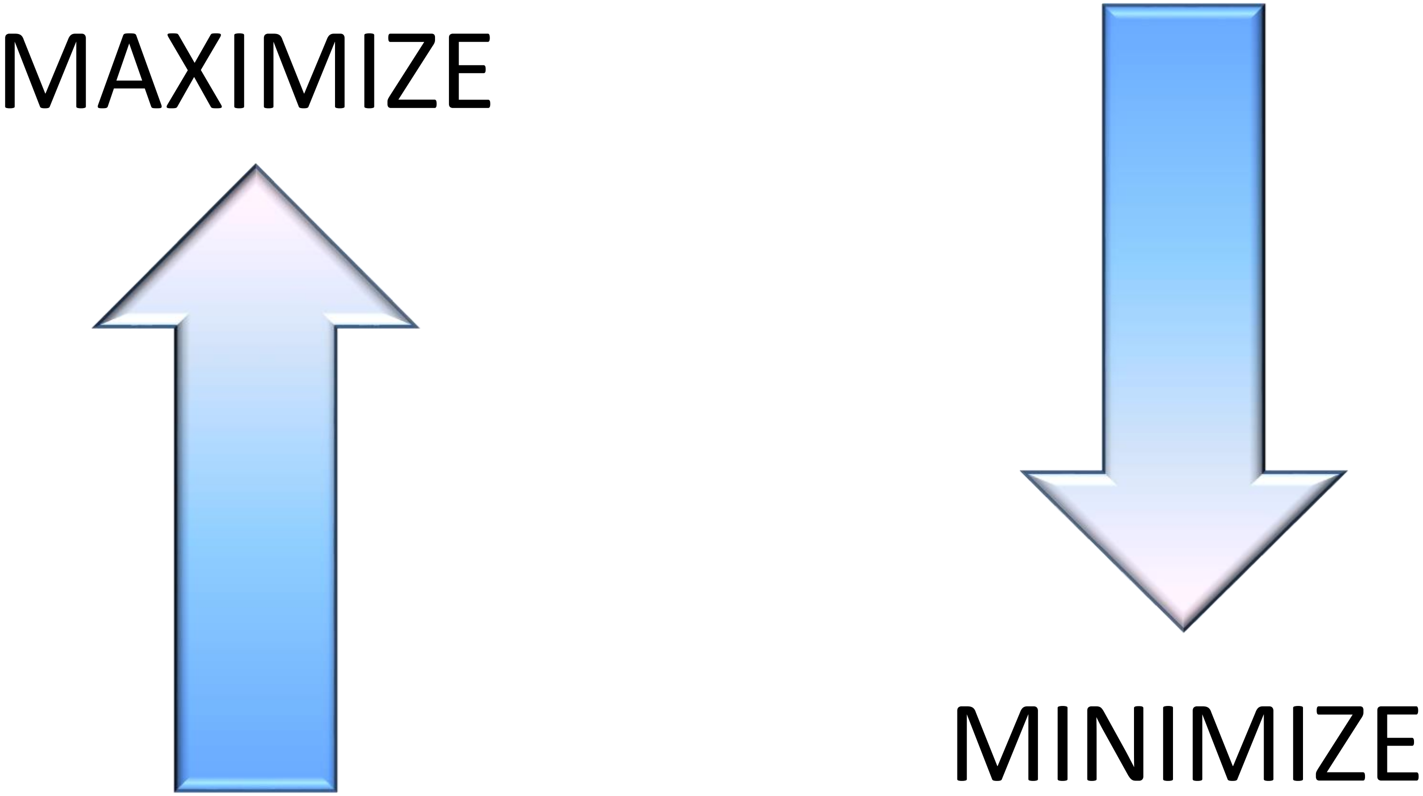

HERE ARE MY CORRECTIONS.
I WANT TO SEE IT AGAIN BEFORE YOU SUBMIT IT.
THE ENDLESS CYCLE?
No!
MAKES CORRECTIONS
MAKES EDITS
HERE'S THE LATEST DRAFT.
JORGE CHAM ©2017
WWW.PHDCOMICS.COM

# What is Optimization ?


INPUTS

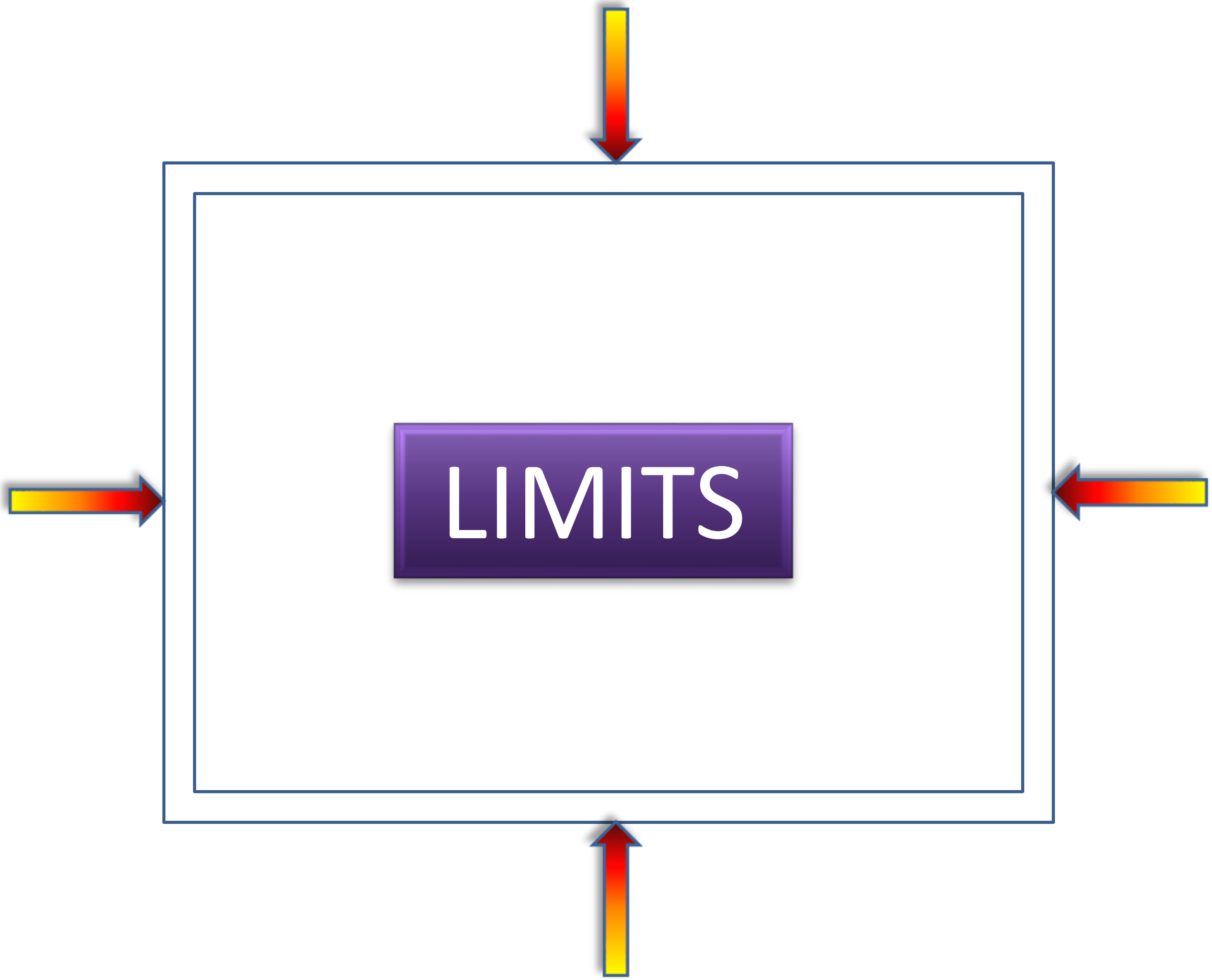
OUTPUTS


ALLOCATE
DESIGN
CONTROL

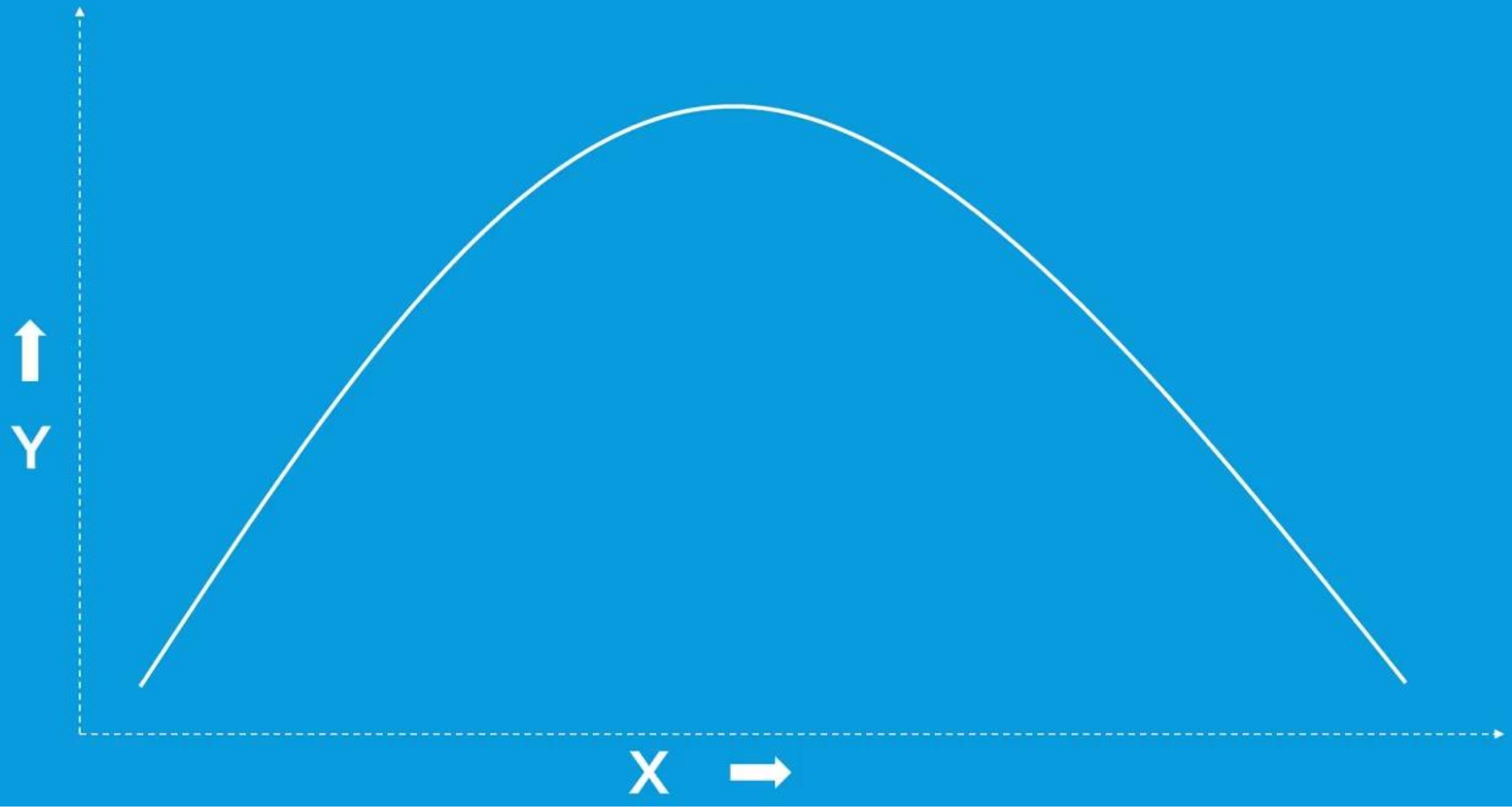


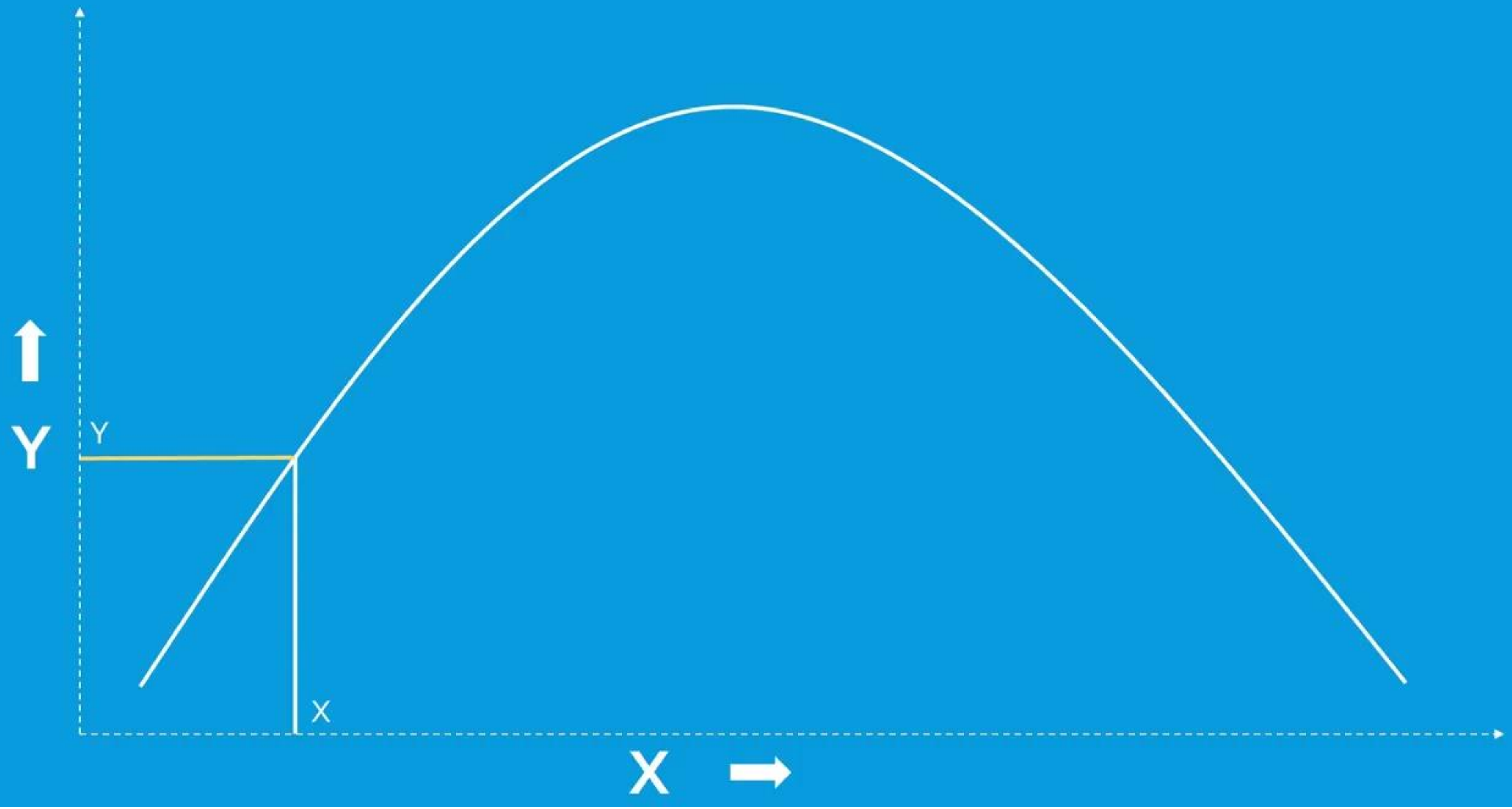
MAXIMIZE

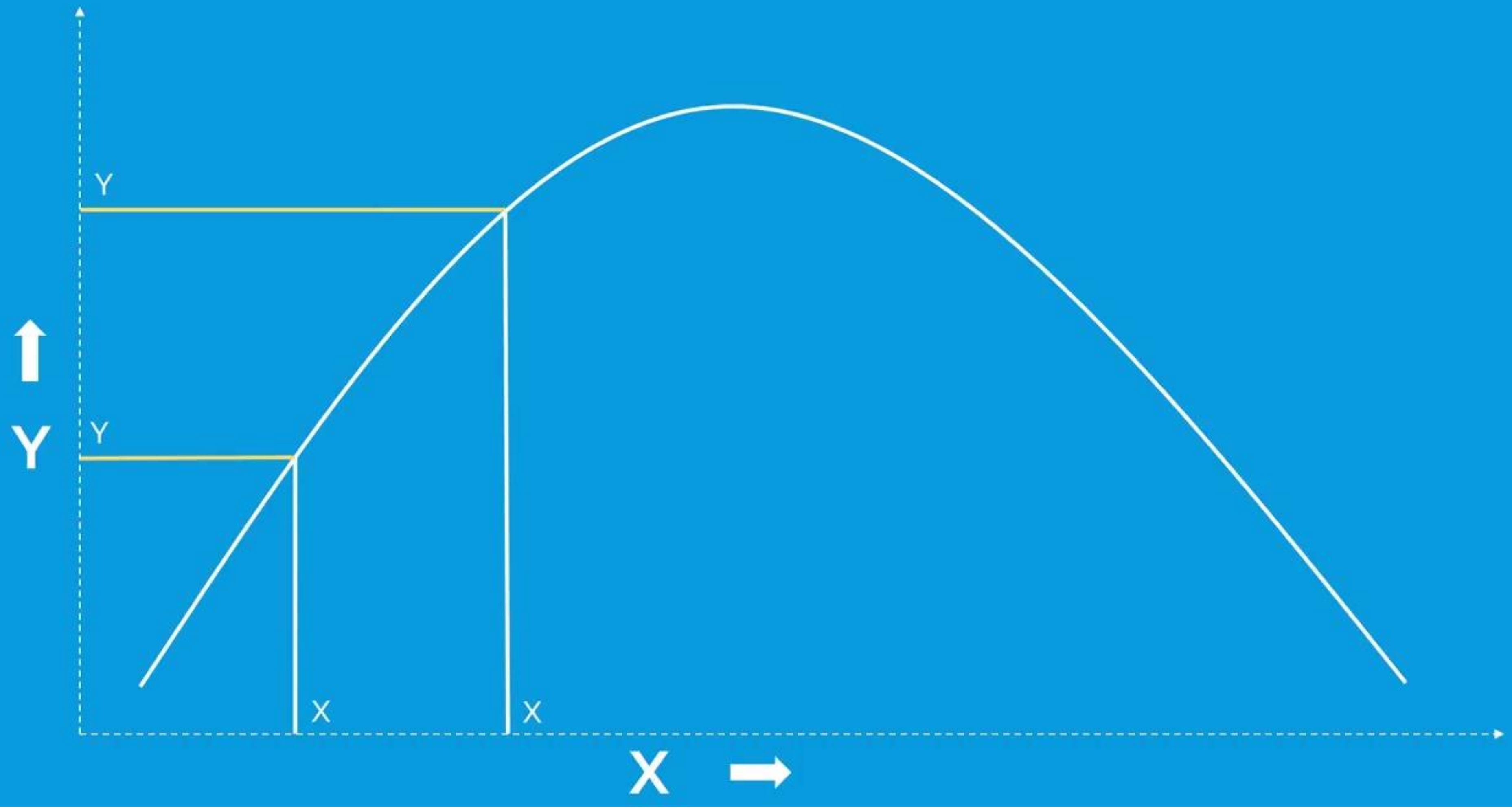
MINIMIZE

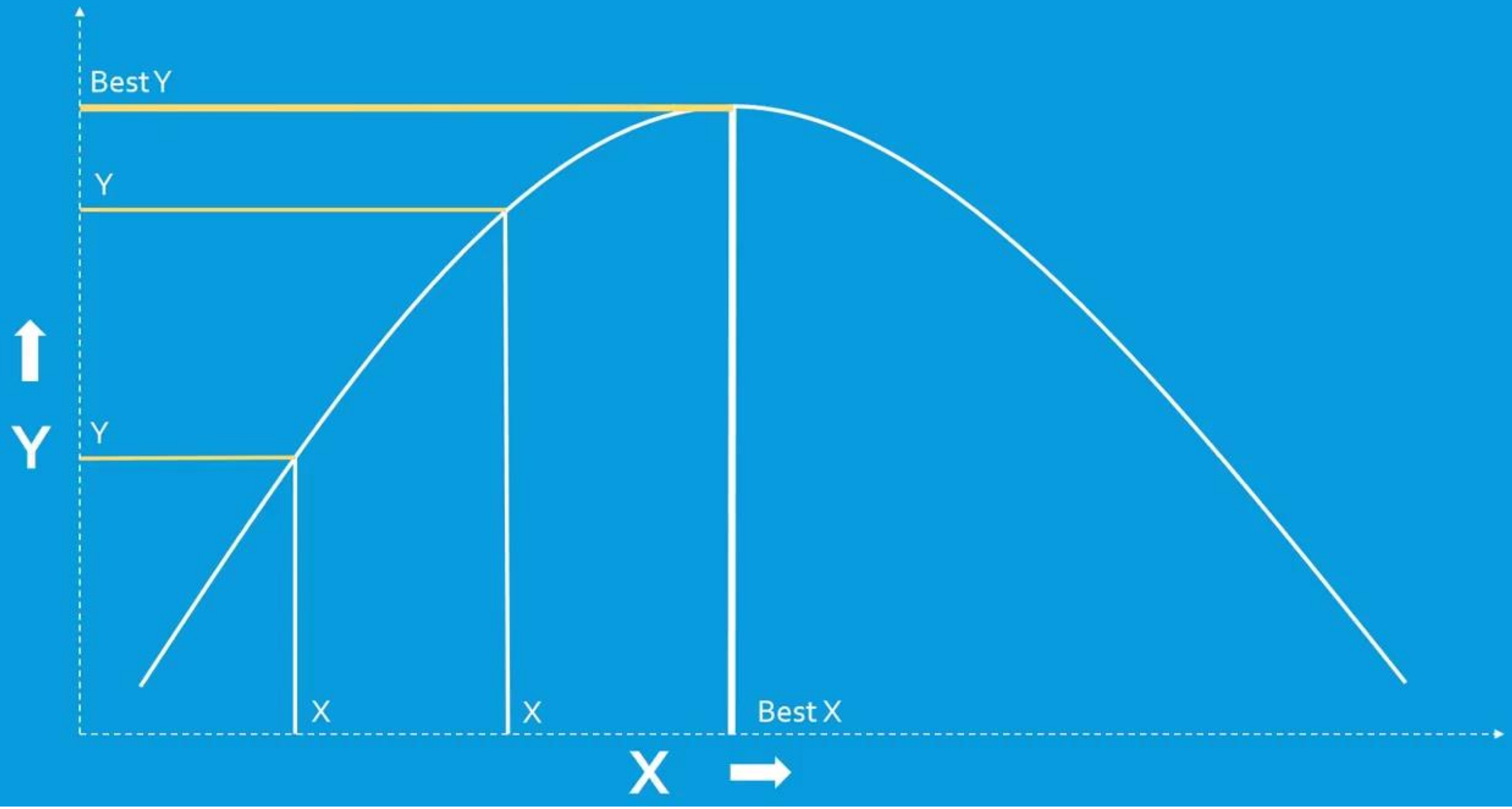


LIMITS


Y
X

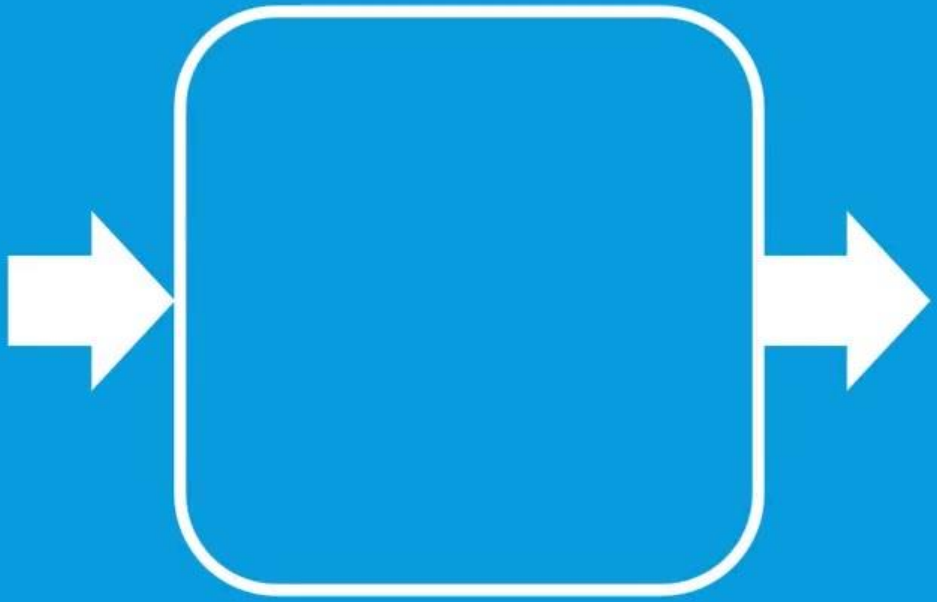

⬆
Y
Y
X
X ➡


Y
Y
Y
X
X
X


Best Y
Y
Y
Y
X
X
Best X
X

# OPTIMIZATION ALGORITHM

# Warehouse Placement

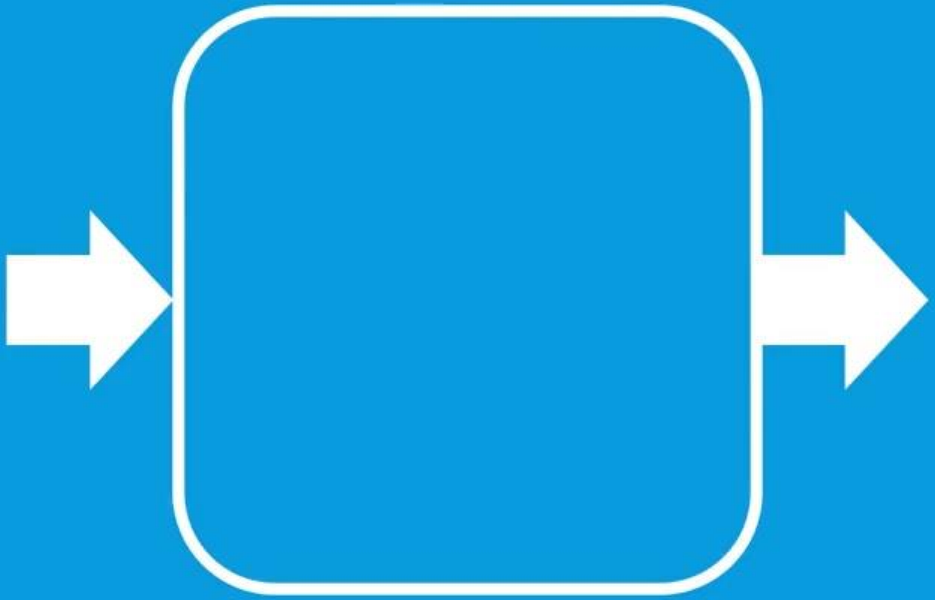


# Warehouse Placement

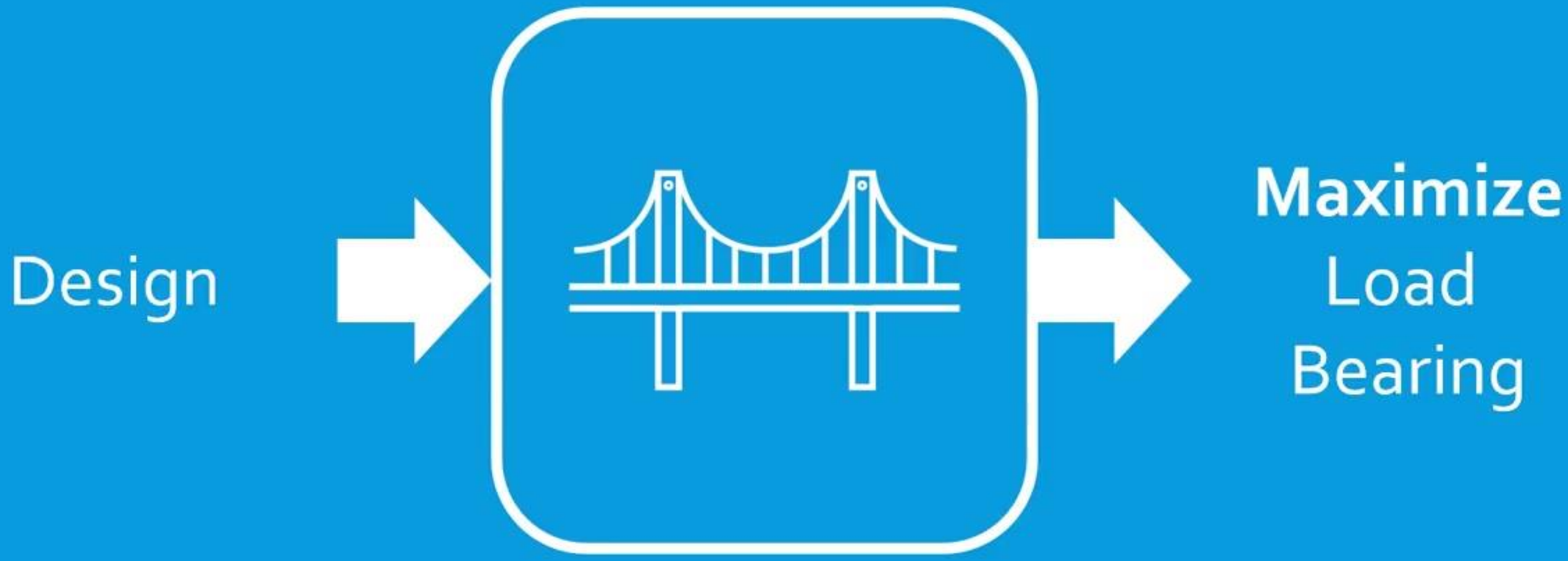
Warehouse Location

**Minimize** Shipment Time

# Bridge Construction



# Bridge Construction



Design

**Maximize** Load Bearing

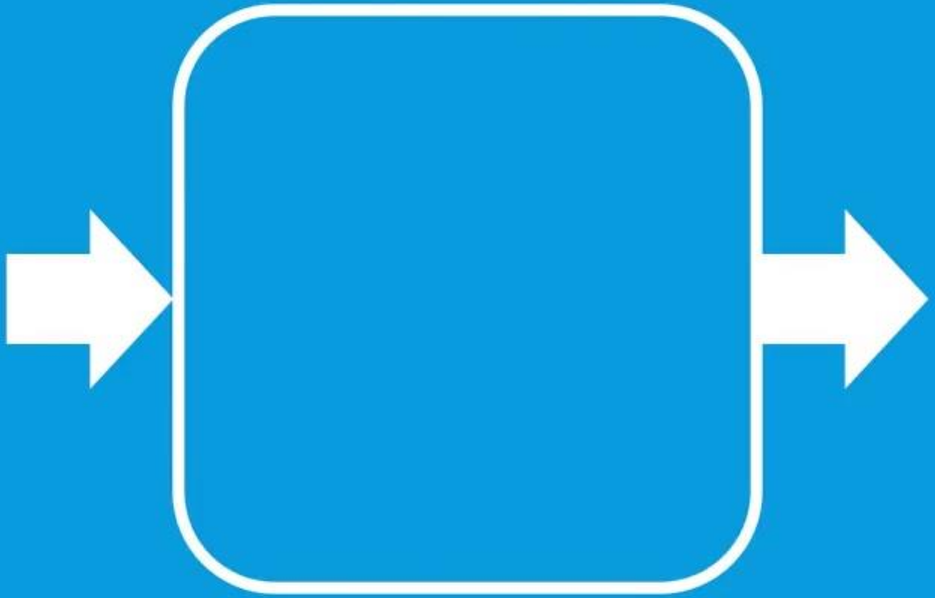
# Strategy Games

# Strategy Games

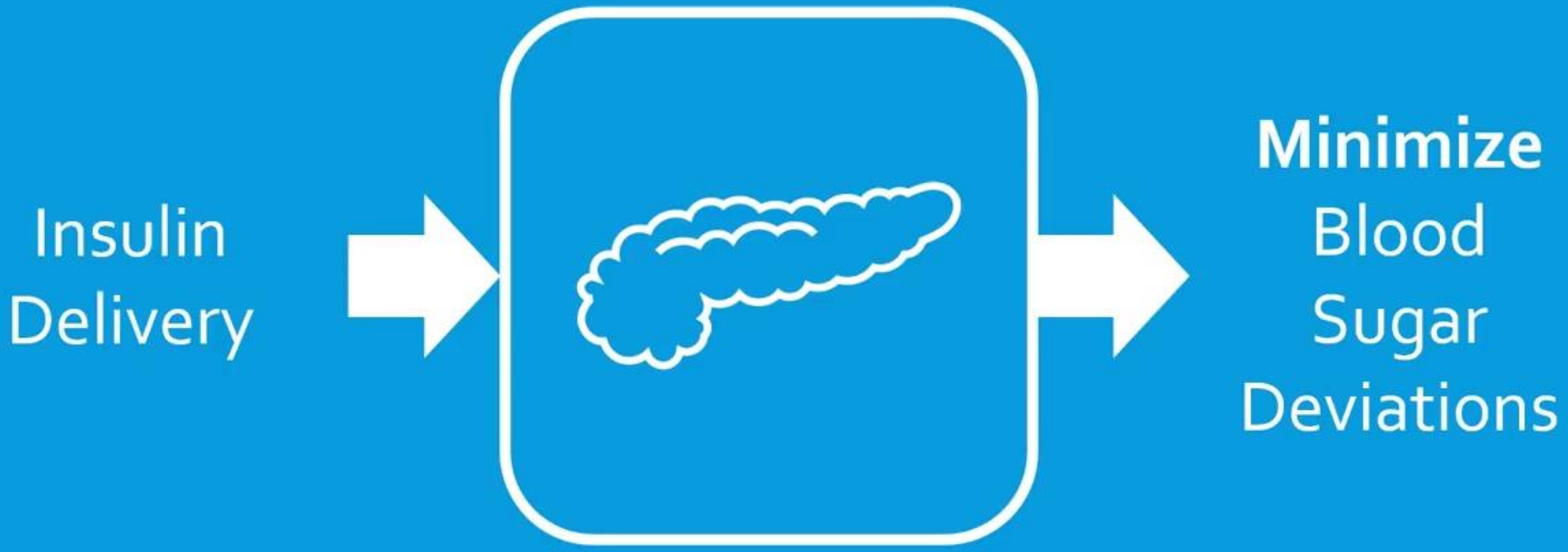
Build Order

**Maximize** Army Strength

# Artificial Pancreas



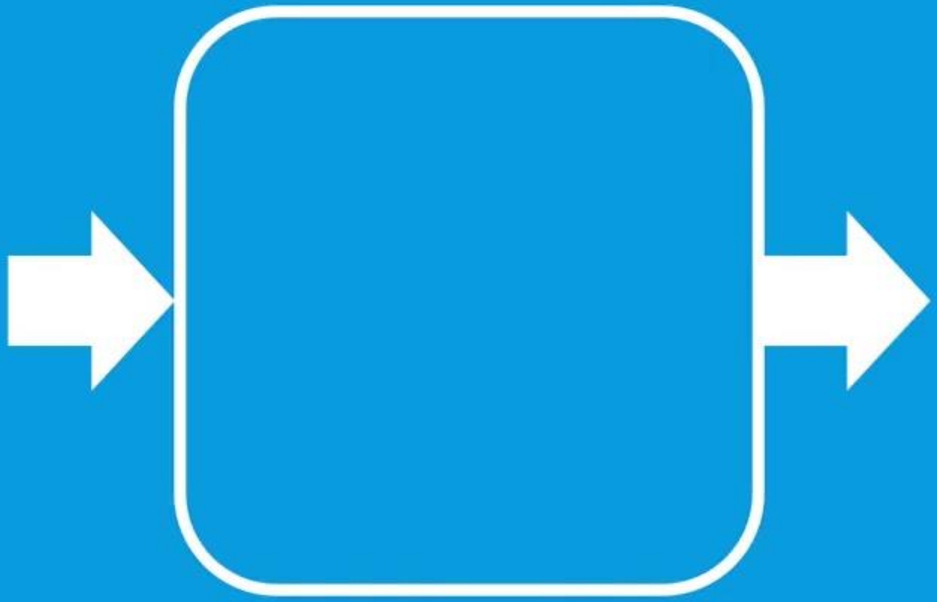
# Artificial Pancreas



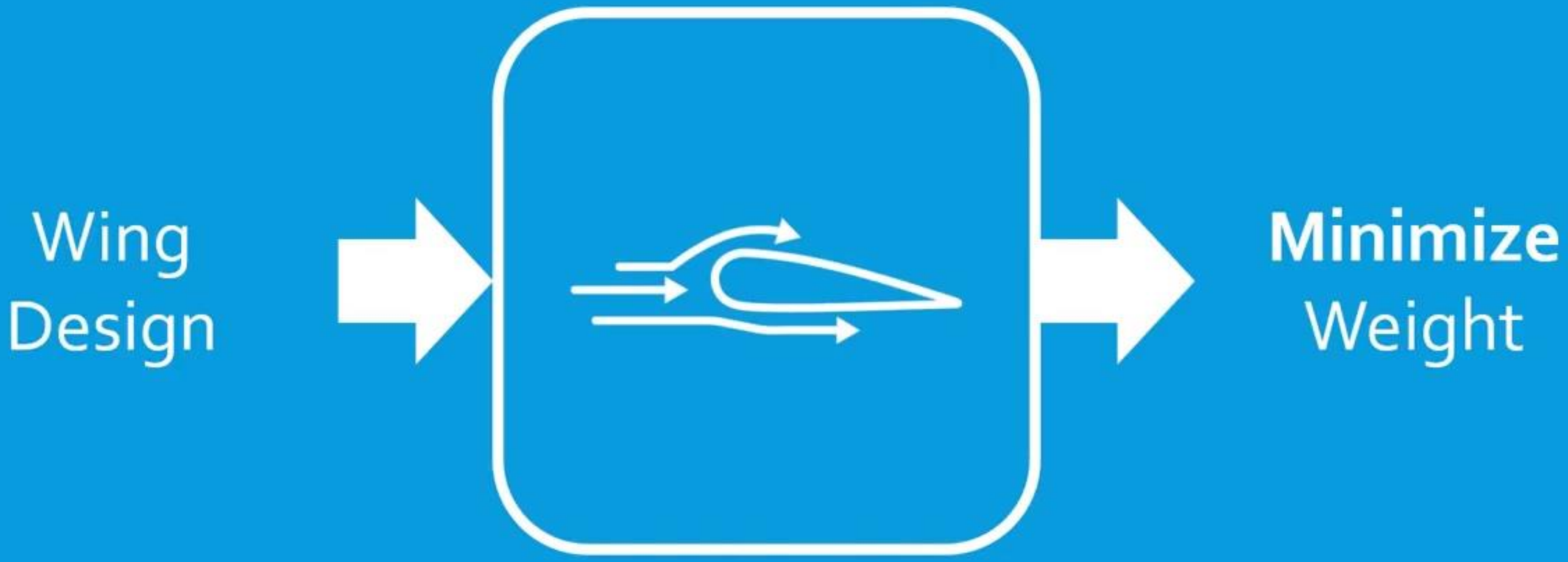
Insulin Delivery

**Minimize** Blood Sugar Deviations

# Airplane Design



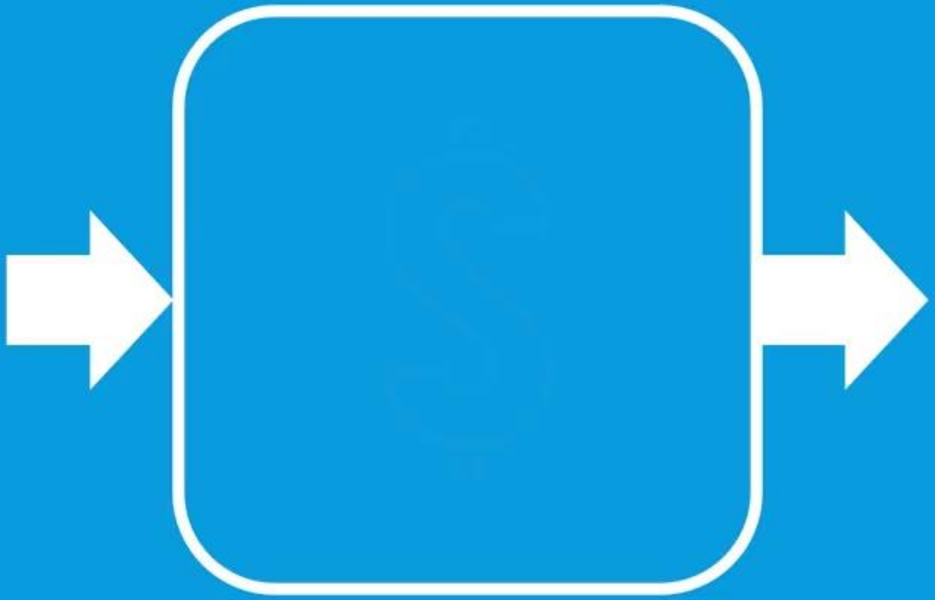
# Airplane Design



Wing Design

**Minimize** Weight

# Stock Market



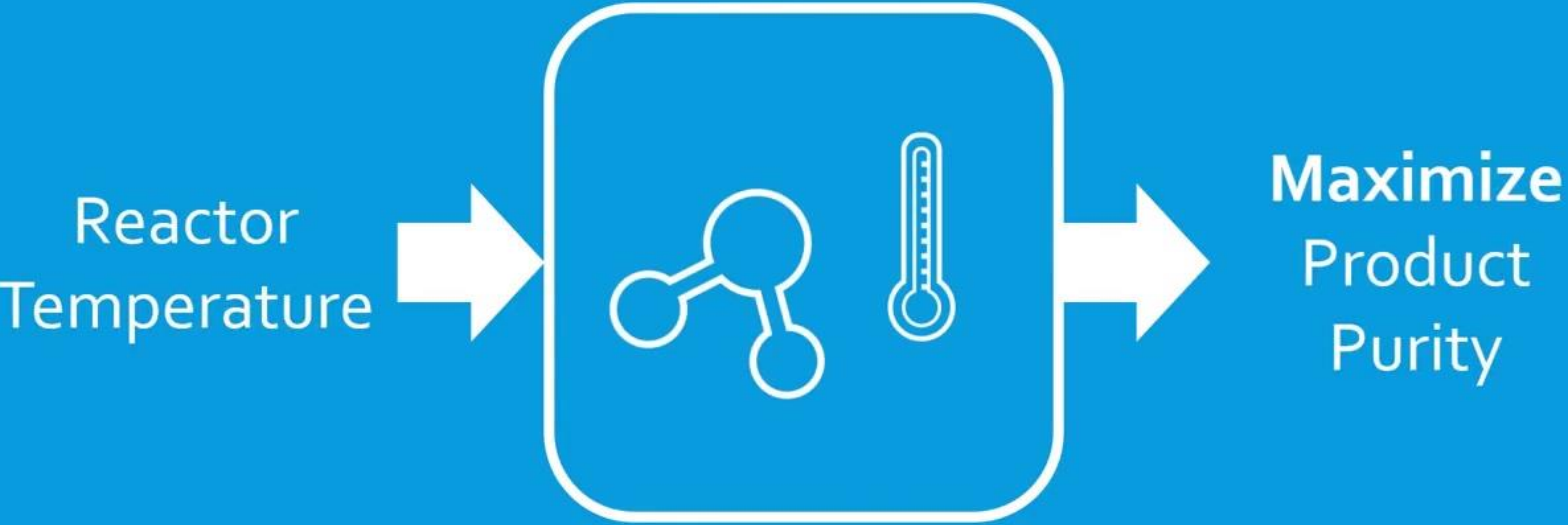
# Stock Market

Stock Portfolio

**Maximize** Returns

# Chemical Reactions

# Chemical Reactions



Reactor Temperature

**Maximize** Product Purity

# Optimization

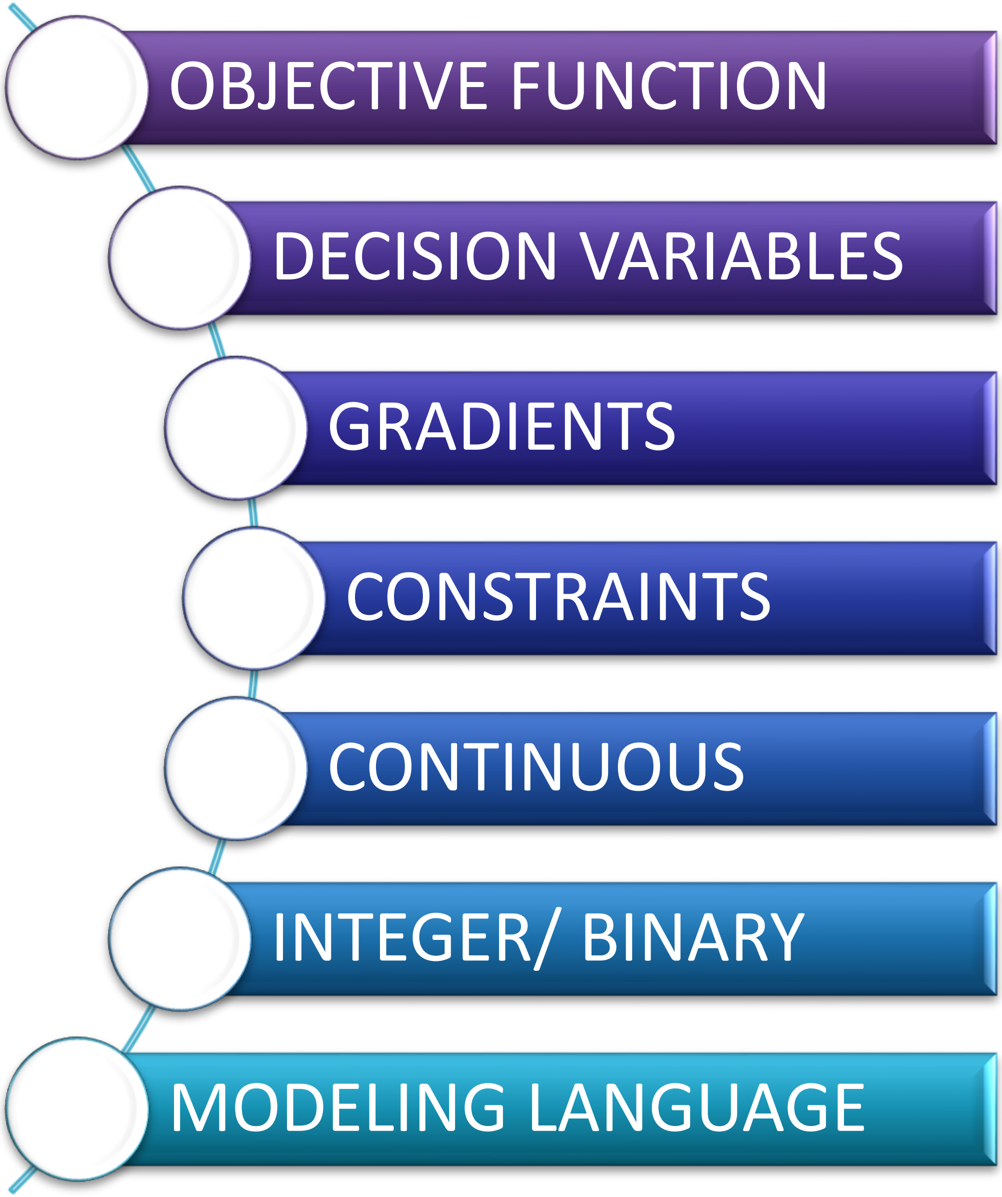
# SUMMARY

1. Helps find the inputs that gives the best outputs

2. Usually requires an optimization algorithm

3. Applicable to many disciplines

# WORDS WHICH CAN DESCRIBE THE IDEA OF OPTIMIZATION



OBJECTIVE FUNCTION

DECISION VARIABLES

GRADIENTS

CONSTRAINTS

CONTINUOUS

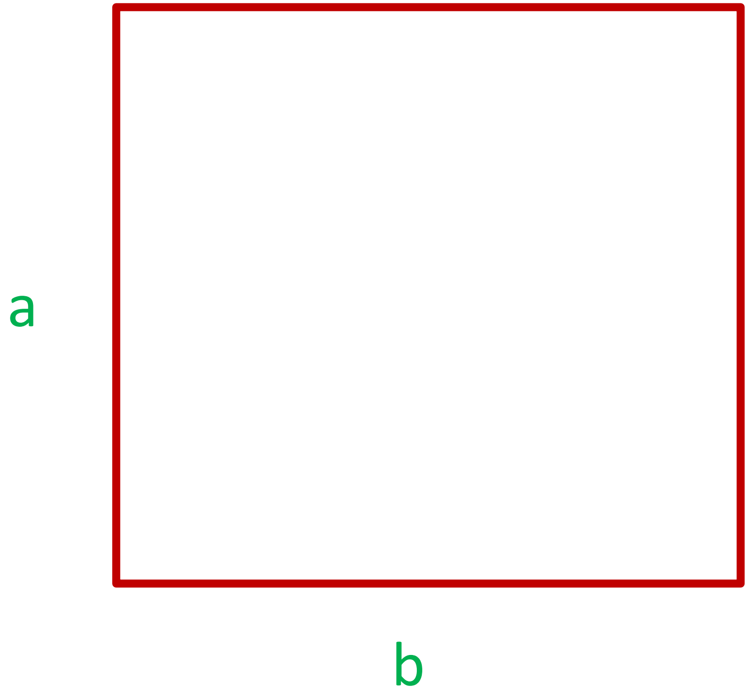
INTEGER/ BINARY

MODELING LANGUAGE

# OBJECTIVE FUNCTION

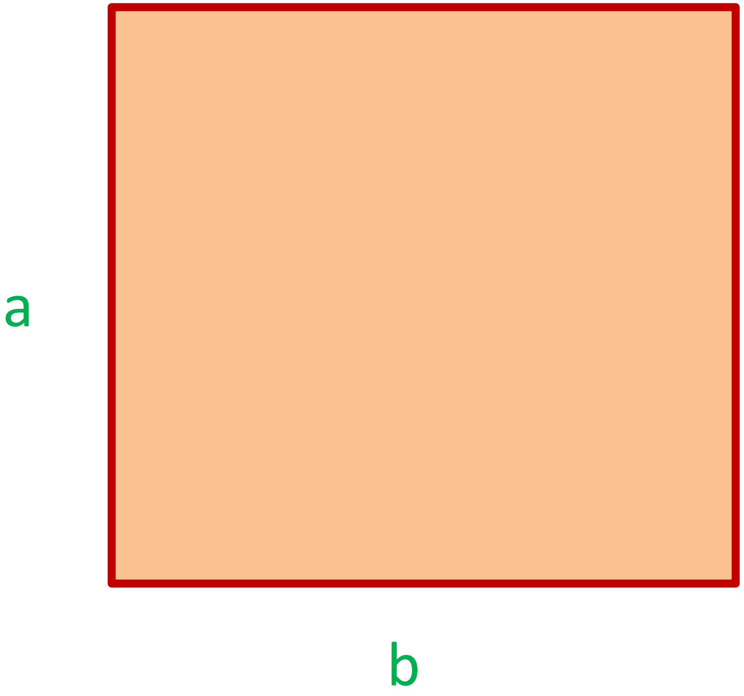
IS THE VALUE WE
ARE TRYING TO OPTIMIZE

# OBJECTIVE FUNCTION



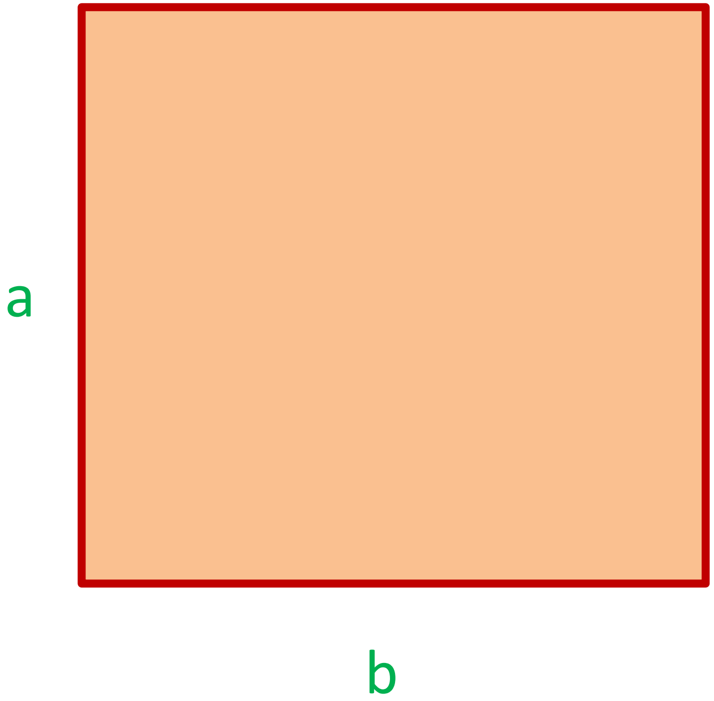
AREA = a x b

# OBJECTIVE FUNCTION

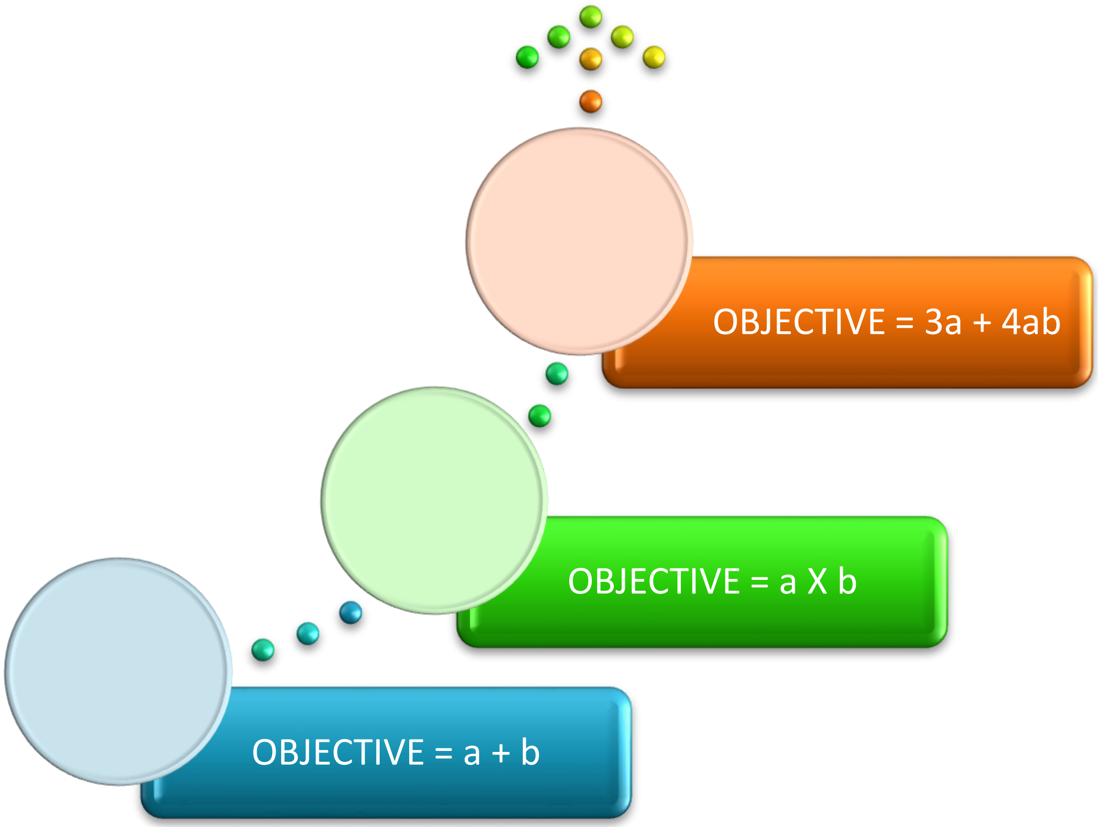


AREA = a x b

# OBJECTIVE FUNCTION



OBJECTIVES CAN BE.......

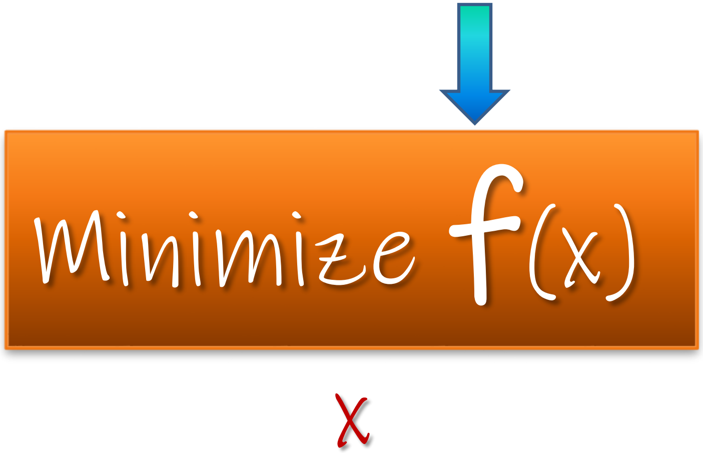


OBJECTIVE = 3a + 4ab

OBJECTIVE = a X b

OBJECTIVE = a + b

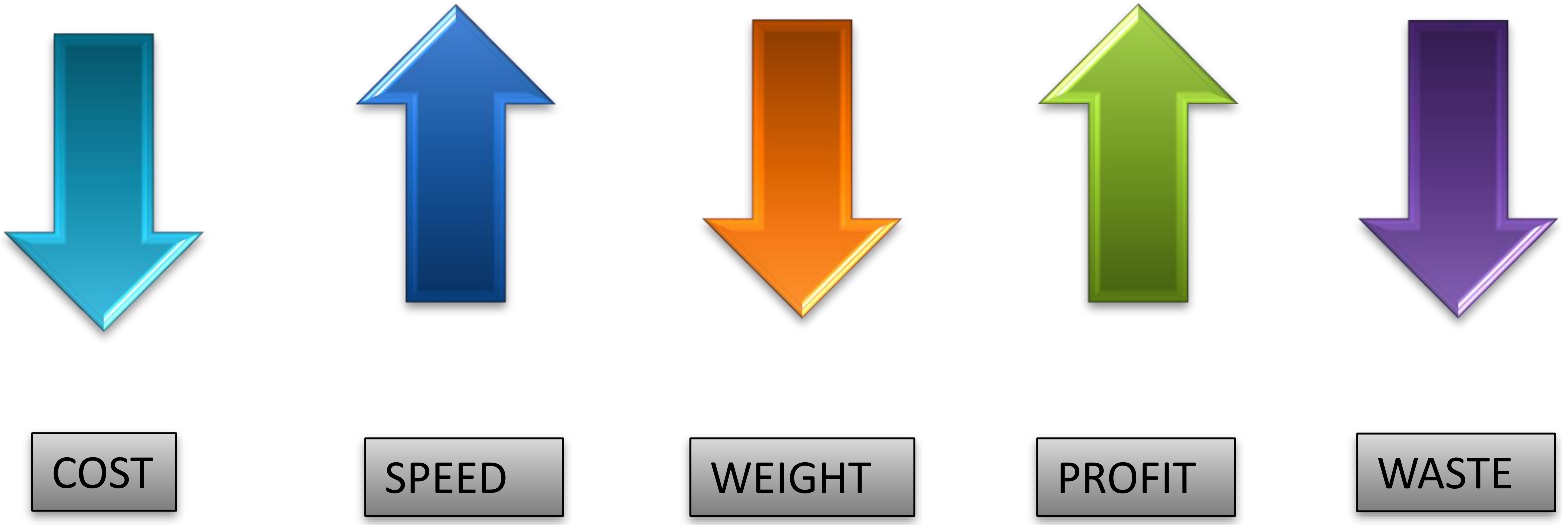
# OBJECTIVE FUNCTION



Minimize $f(x)$

$x$

# OBJECTIVE FUNCTION



COST

SPEED

WEIGHT

PROFIT

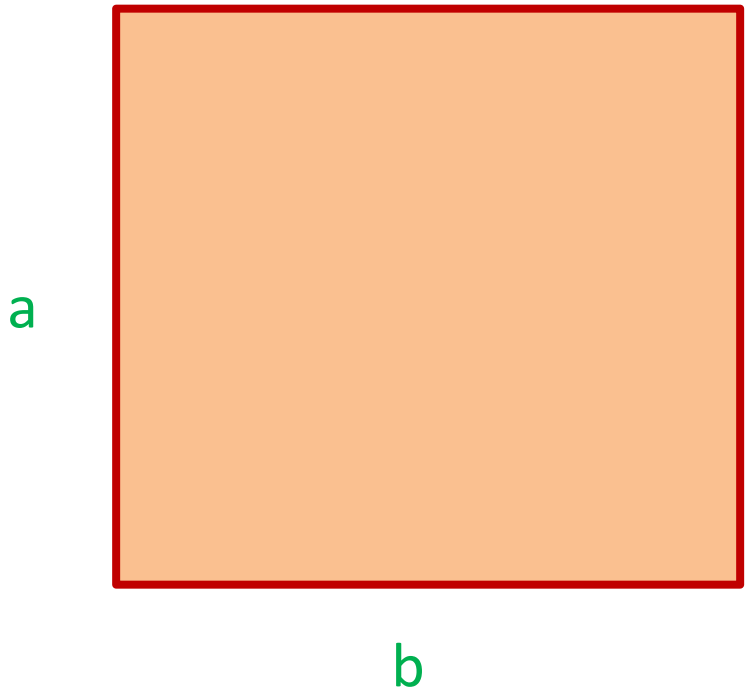
WASTE

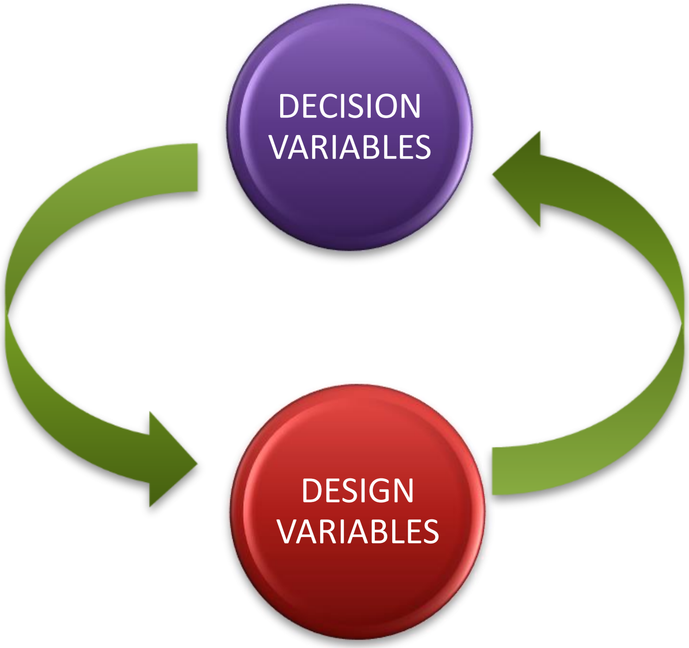
# DECISION VARIABLES

ARE THE VALUES
THE OPTIMIZER CAN CHANGE

# DECISION VARIABLES

AREA = a x b





# DECISION VARIABLES

Minimize $f(x)$

# DECISION VARIABLES

Minimize f(x1, X2, X3……….)

# SUMMARY

OBJECTIVE FUNCTION :

VALUE WHICH WE ARE TRYING TO OPTIMIZE

EITHER MINIMIZED OR MAXIMIZED

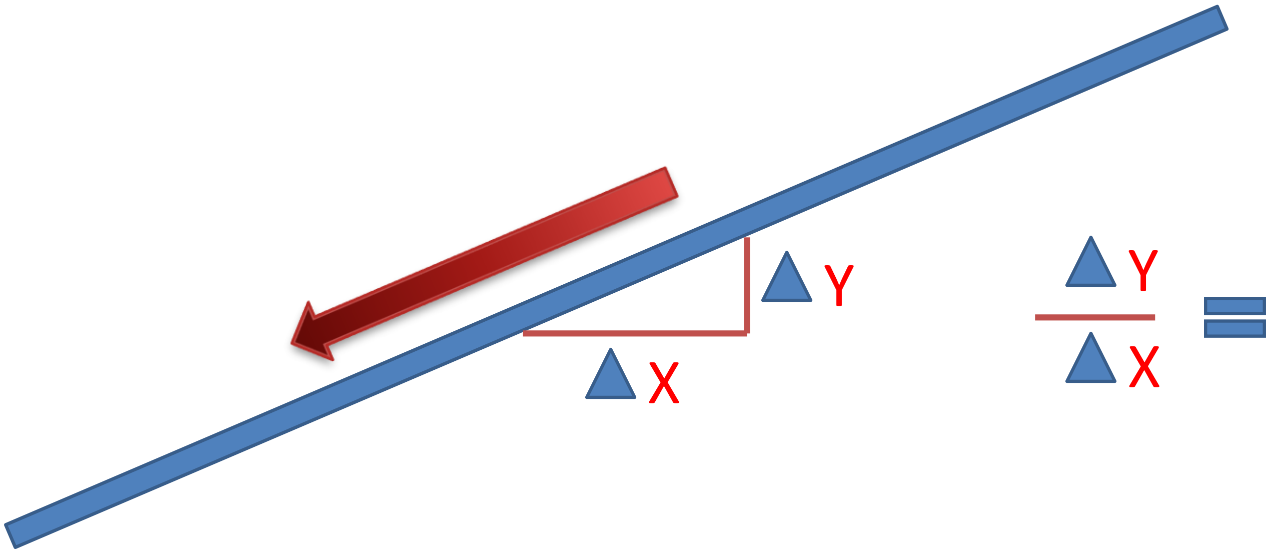
DECISION VARIABLES :

THE VALUES THAT OPTIMIZER CAN CHANGE

# GRADIENTS

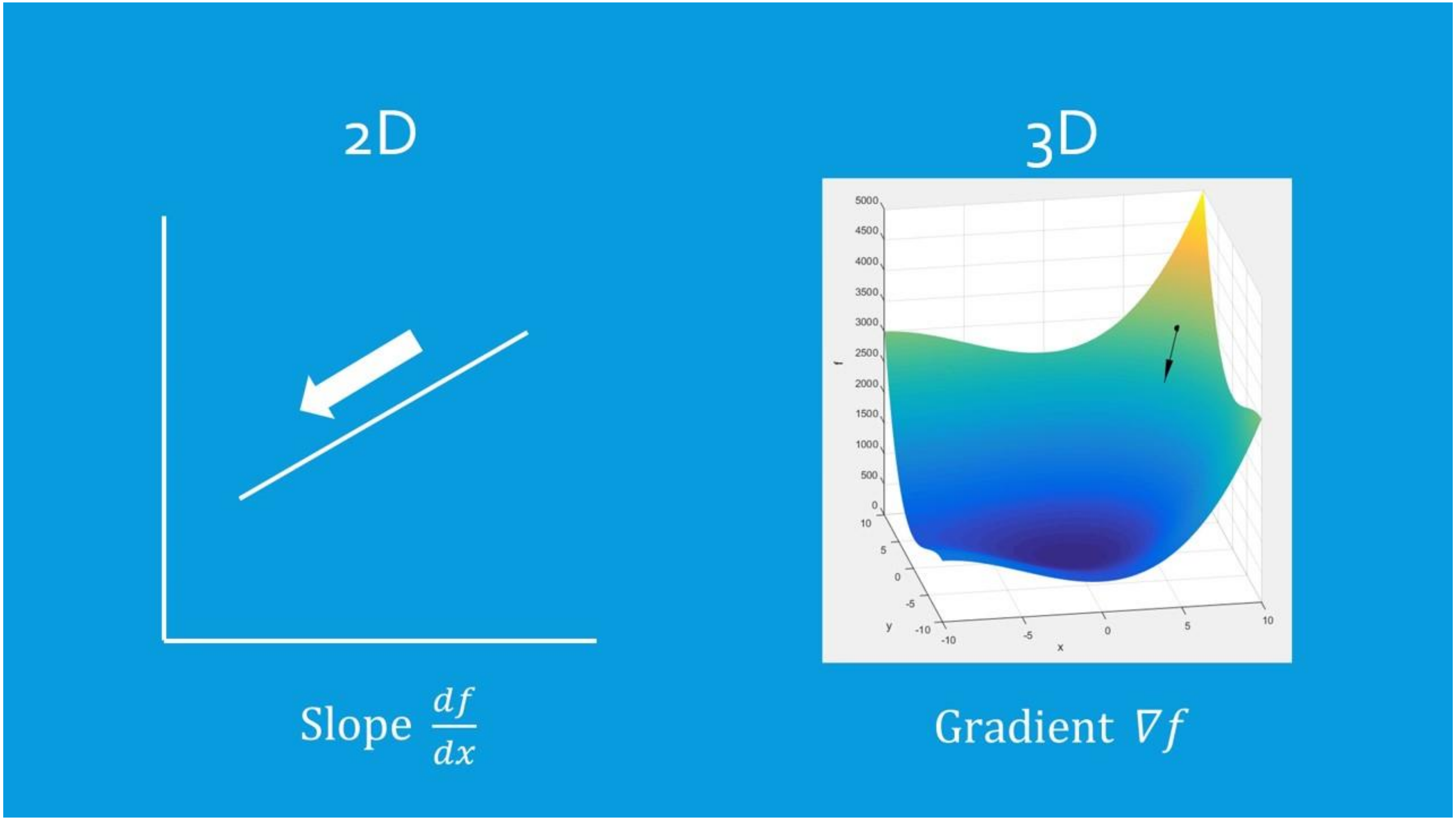
## SLOPE OF THE FUNCTION

# GRADIENTS



$$\frac{\Delta Y}{\Delta X} = \text{SLOPE}$$

# GRADIENTS



2D

Slope $\frac{df}{dx}$

3D

Gradient $\nabla f$

# GRADIENTS

NUMERICAL

ANALYTICAL

AUTOMATIC

## DIFFERENTIATION

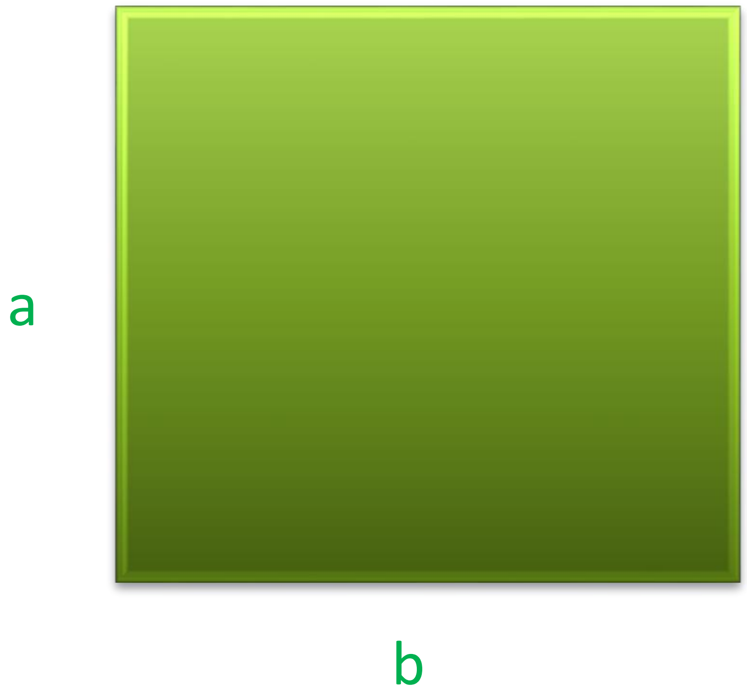
$\nabla f$

# CONSTRAINTS

WHERE THE OPTIMIZER CANNOT GO

# CONSTRAINTS

$\text{AREA} = a \times b$



INEQUALITY CONSTRAINT

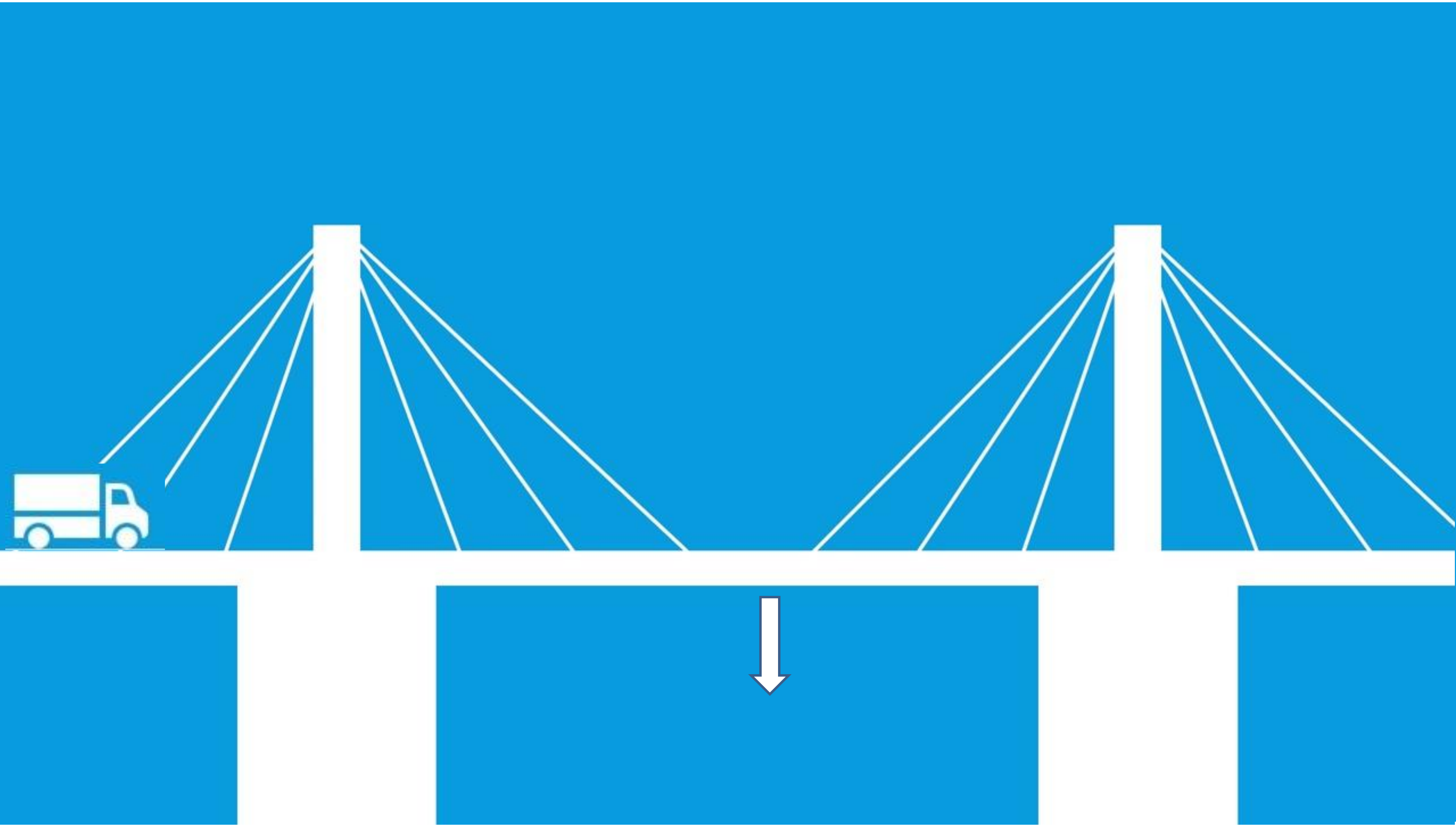
$a \times b < = 5$

EQUALITY CONSTRAINT

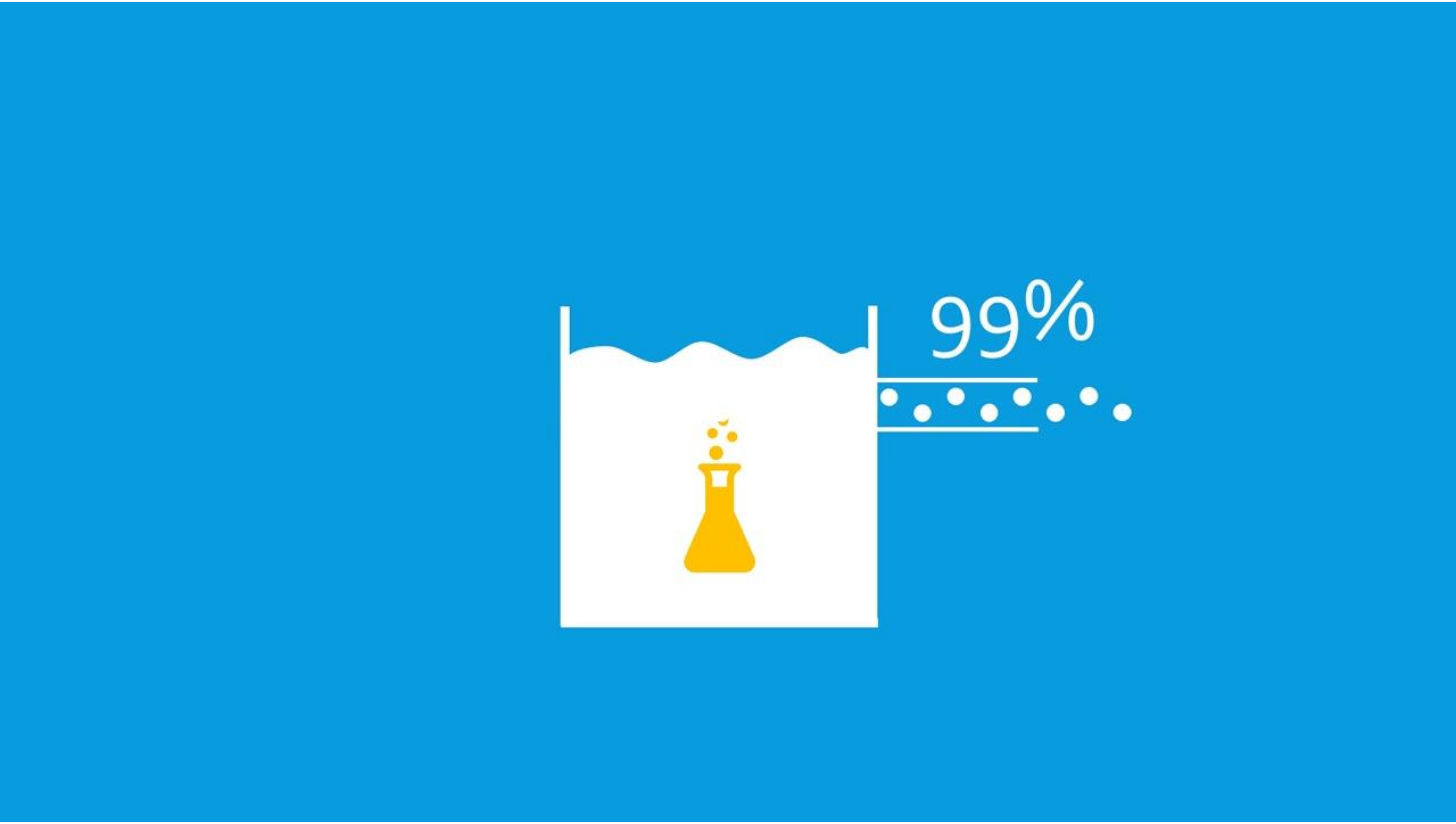
$a + b = 5$

$$\frac{ab}{a + b} = 3$$

# CONSTRAINTS



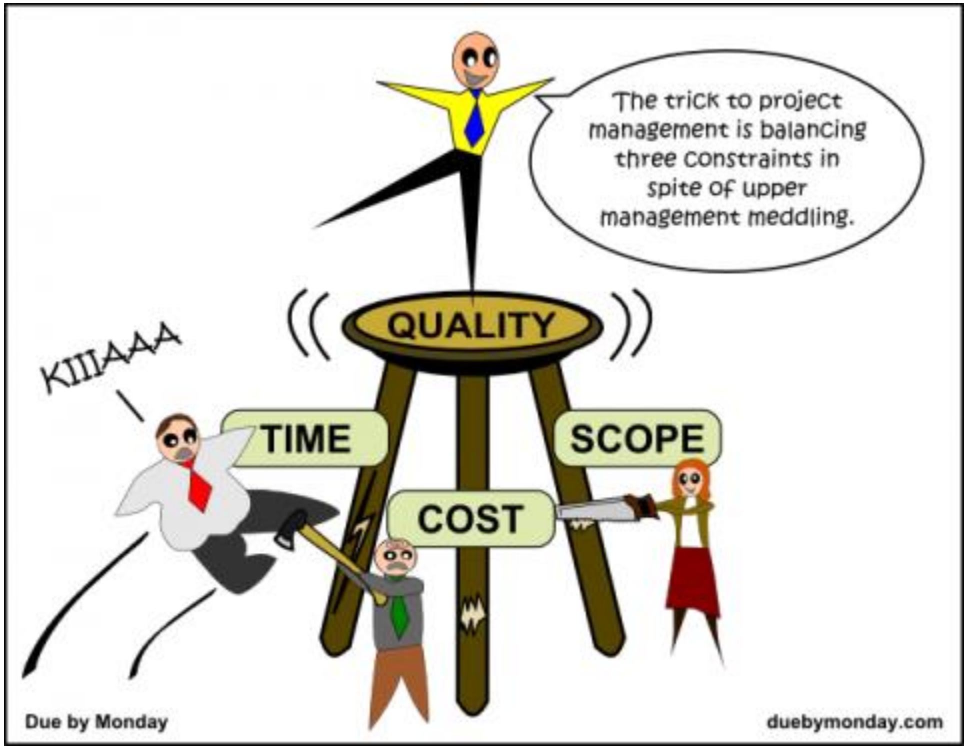
# CONSTRAINTS



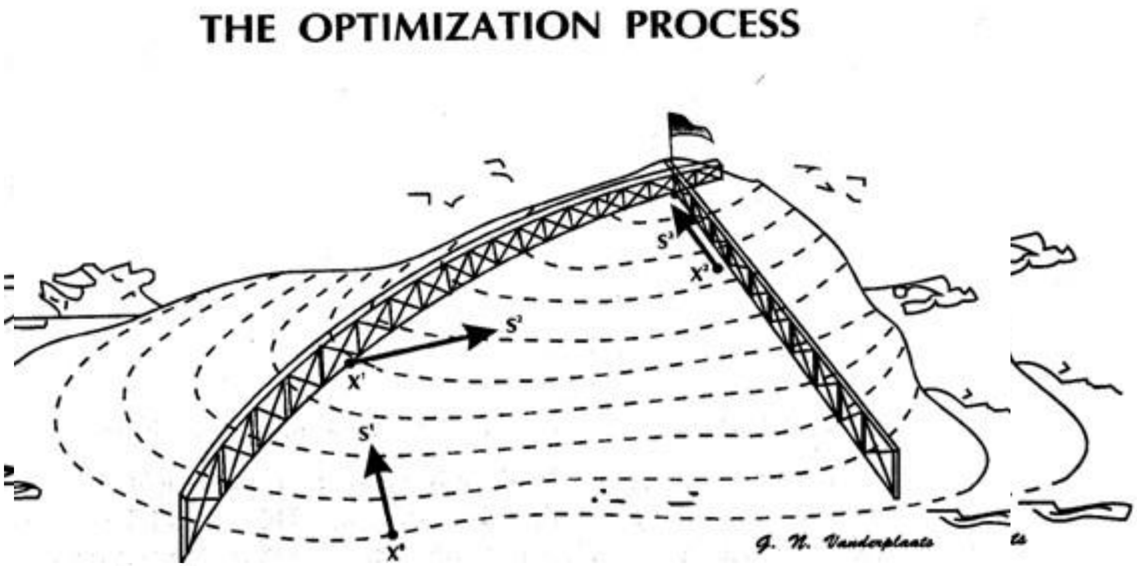
# CONSTRAINTS

"There were some budget constraints on that one."

**BUDGET CONSTRAINT**

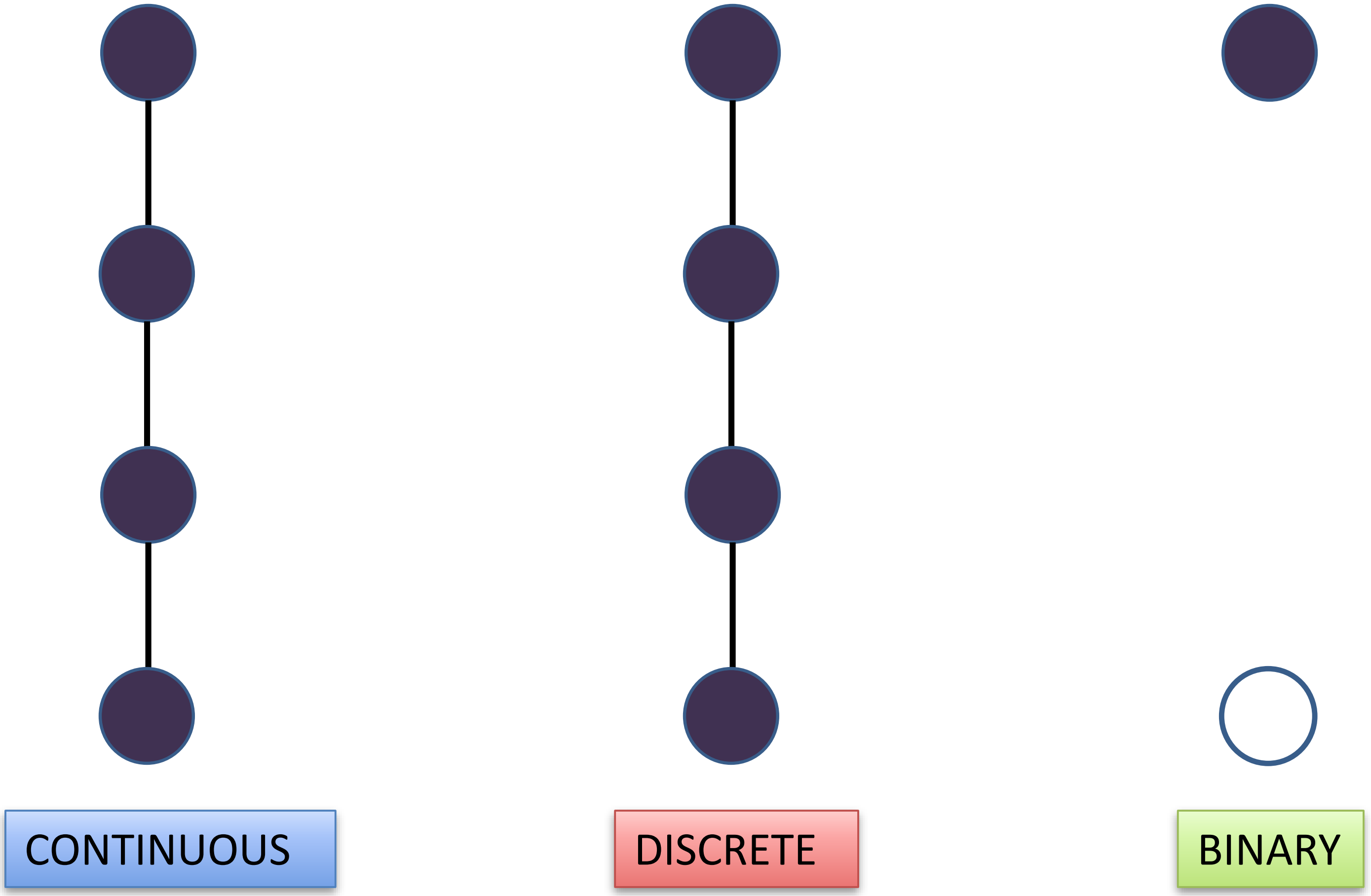

The trick to project management is balancing three constraints in spite of upper management meddling.
QUALITY
KIIIAAA
TIME
SCOPE
COST
Due by Monday
duebymonday.com

# CONSTRAINTS



THE OPTIMIZATION PROCESS

# CONTINUOUS VARIABLES



CONTINUOUS

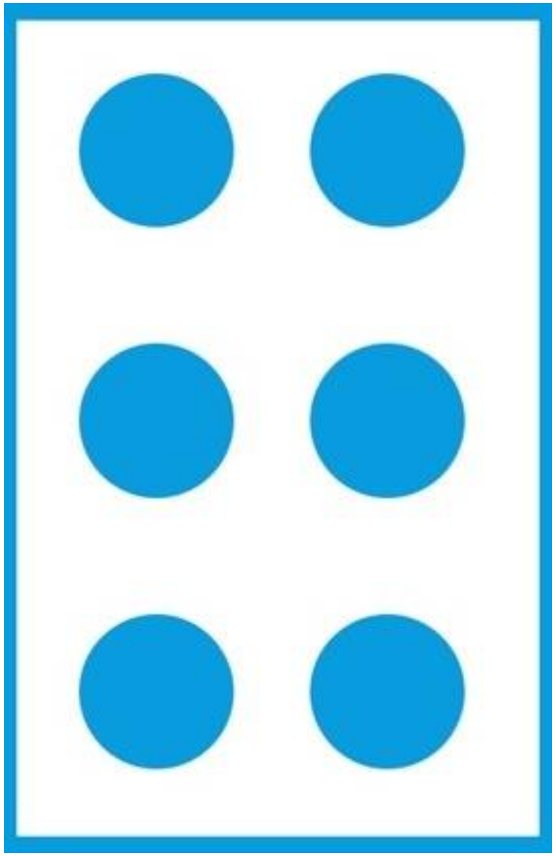
DISCRETE

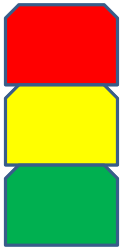
BINARY

# CONTINUOUS VARIABLES

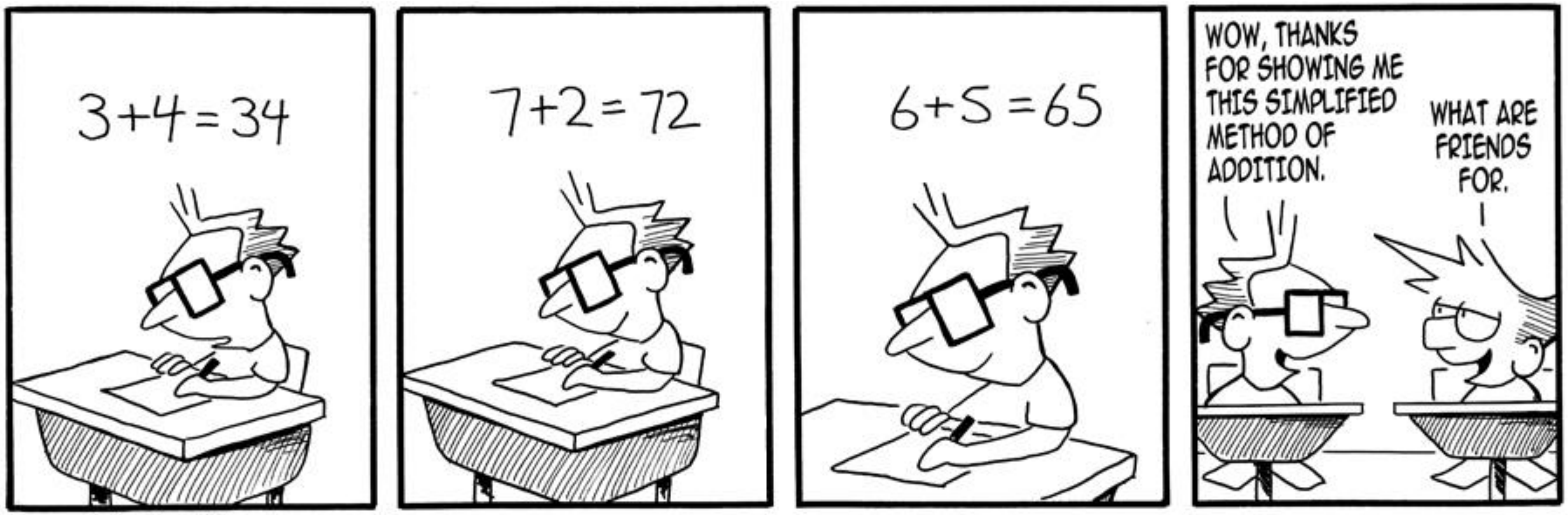
20 KMPH
50 KMPH

# DISCRETE VARIABLES





# BINARY VARIABLES


3+4=34
7+2=72
6+5=65
WOW, THANKS FOR SHOWING ME THIS SIMPLIFIED METHOD OF ADDITION.
WHAT ARE FRIENDS FOR.

FLOYDCOMICS.COM

# OBTAINING DERIVATIVES

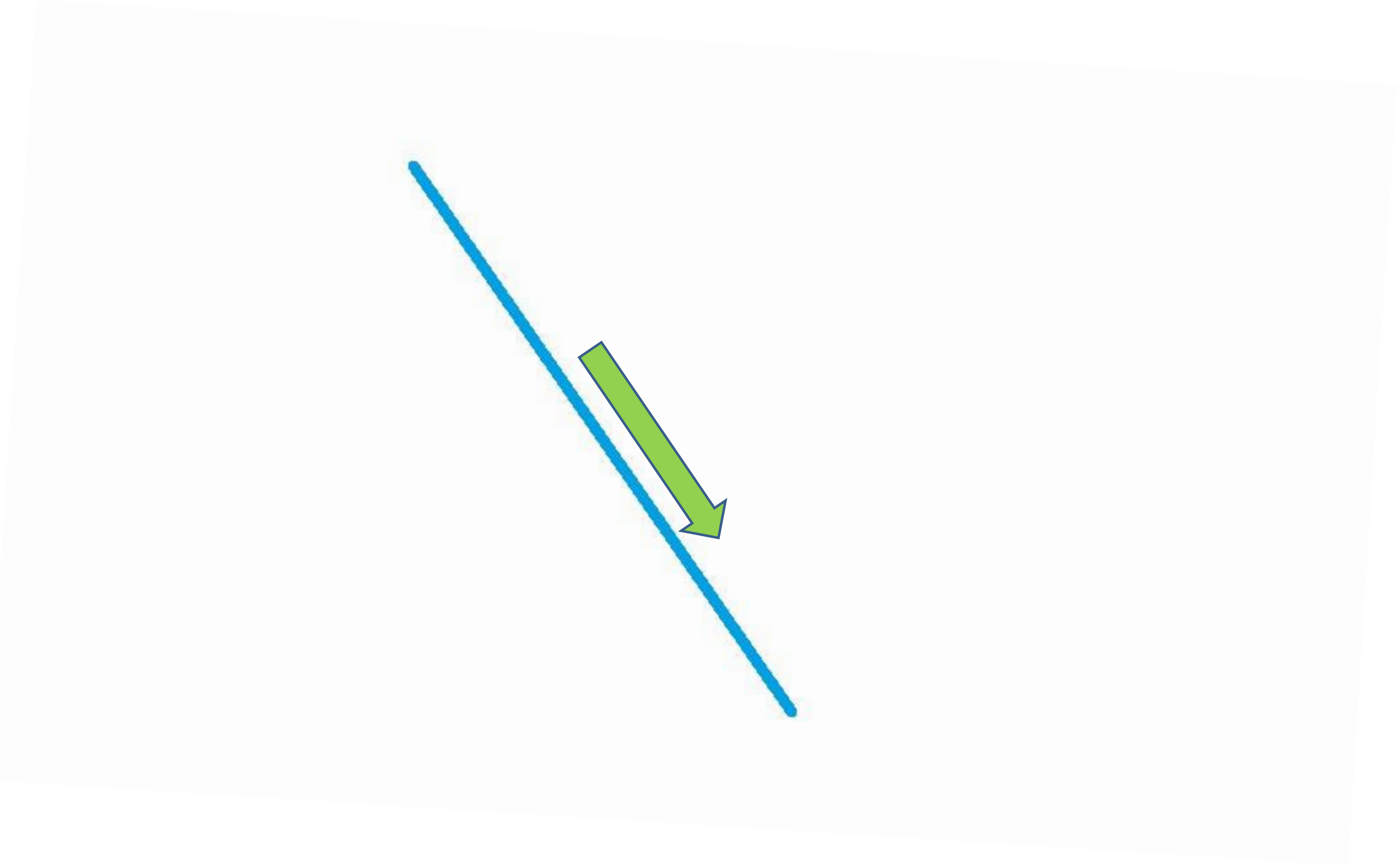
SYMBOLIC

NUMERICAL

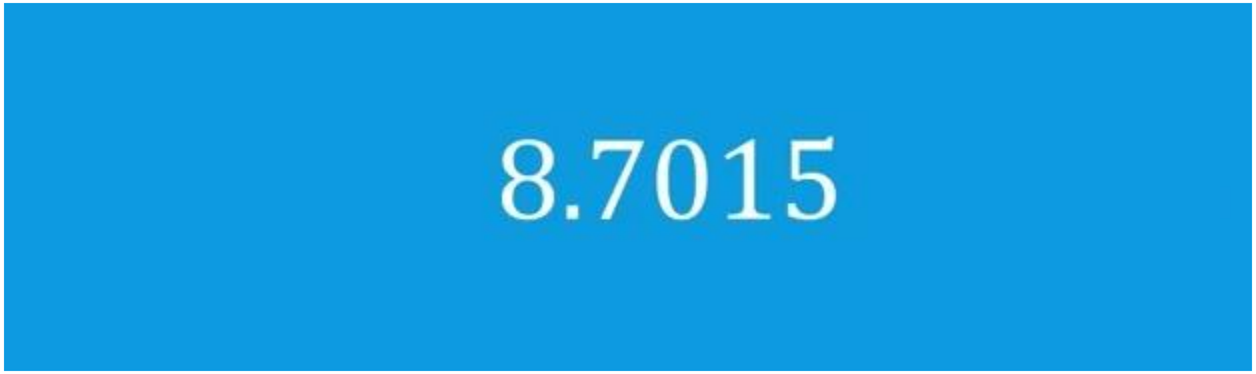
AUTOMATIC

DIFFERENTIATION

# OBTAINING DERIVATIVES



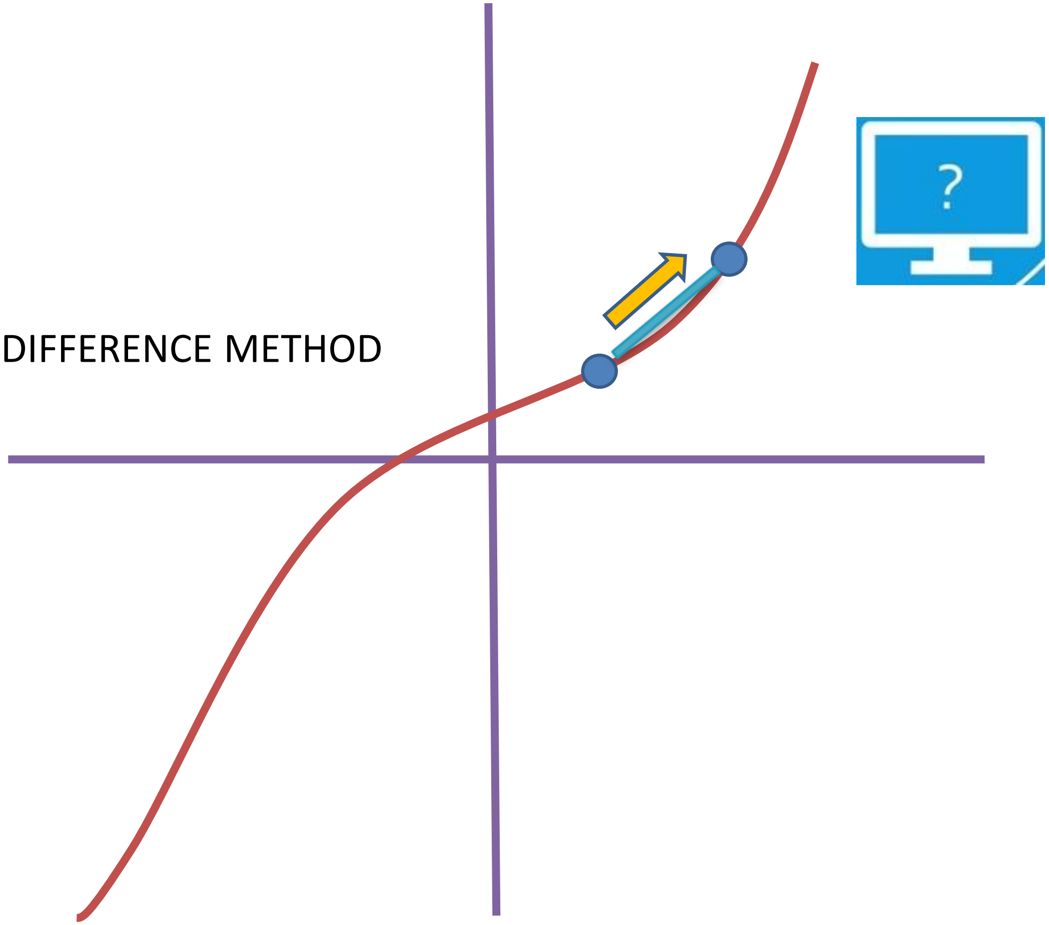
# SYMBOLIC DIFFERENTIATION



```
8.7015
```

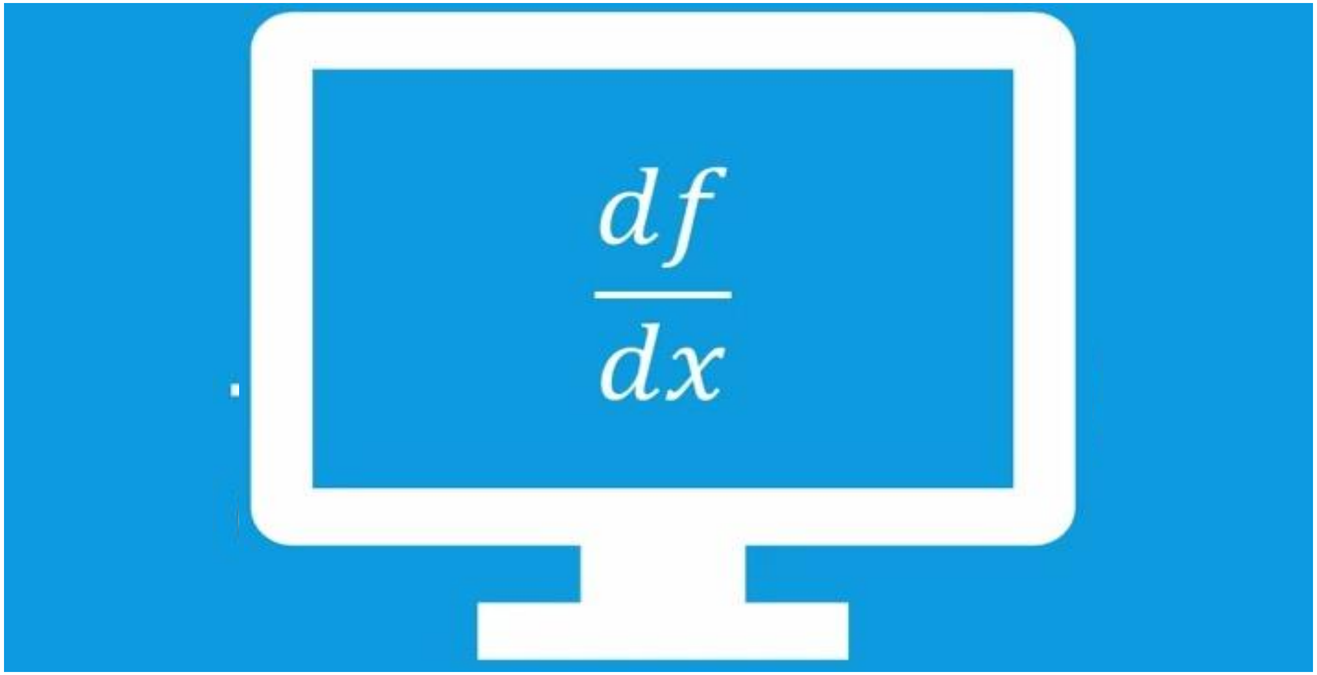
# NUMERICAL DIFFERENTIATION

FINITE DIFFERENCE METHOD

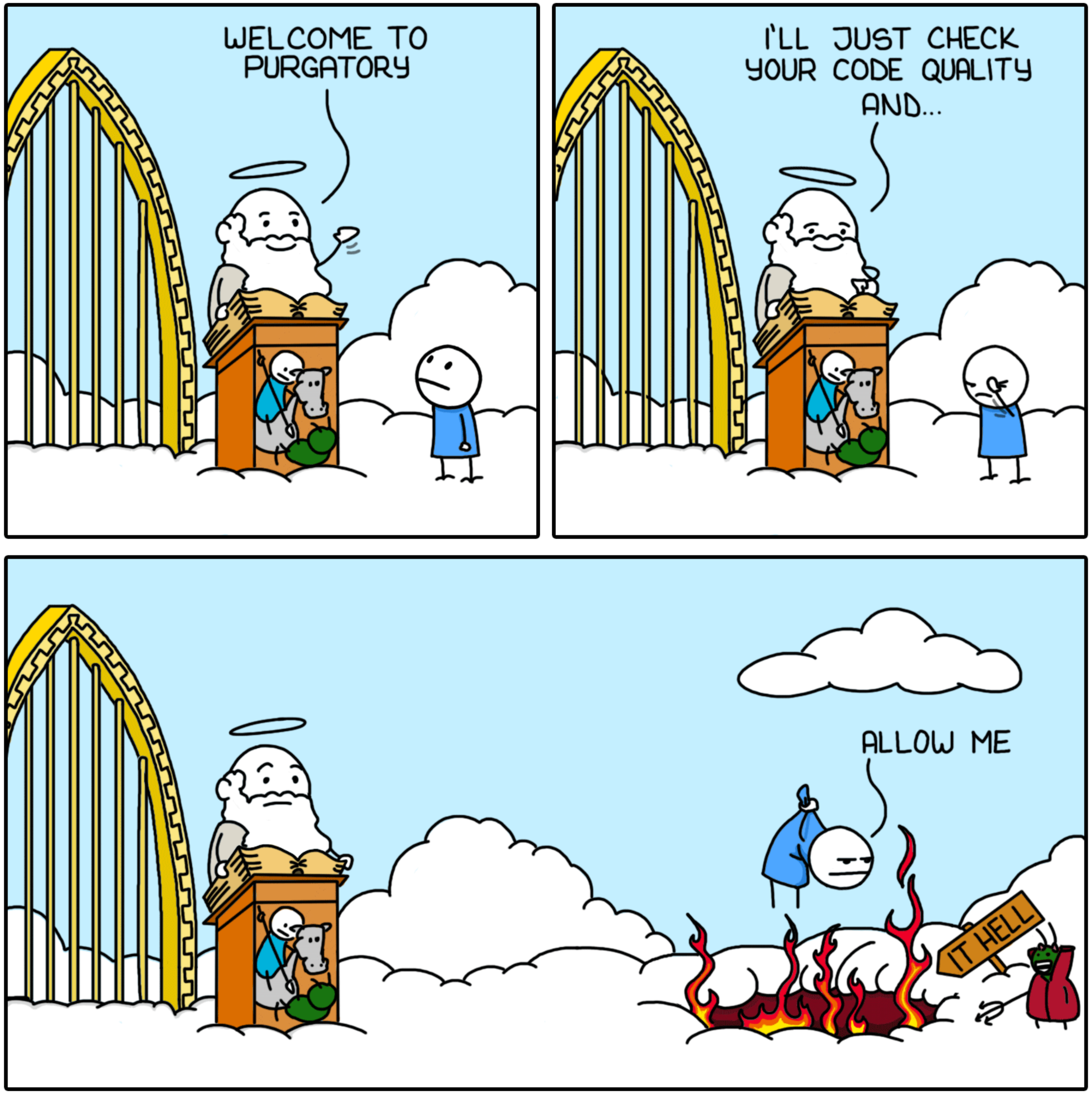


# AUTOMATIC DIFFERENTIATION

## SIMILAR TO SYMBOLIC DIFFERENTIATION



# LAST PUSH



MONKEYUSER.COM

# SUMMARY

SYMBOLIC DIFFERENTIATION:

- PROBLEM INSIGHT
- SLOW

NUMERICAL DIFFERENTIATION:

- EASY IMPLEMENTATION
- INACCURATE

AUTOMATIC DIFFERENTIATION:

- ACCURATE
- MORE DIFFICULT TO IMPLEMENT

# GRADIENT

GRADIENT
BASED
ALGORITHMS

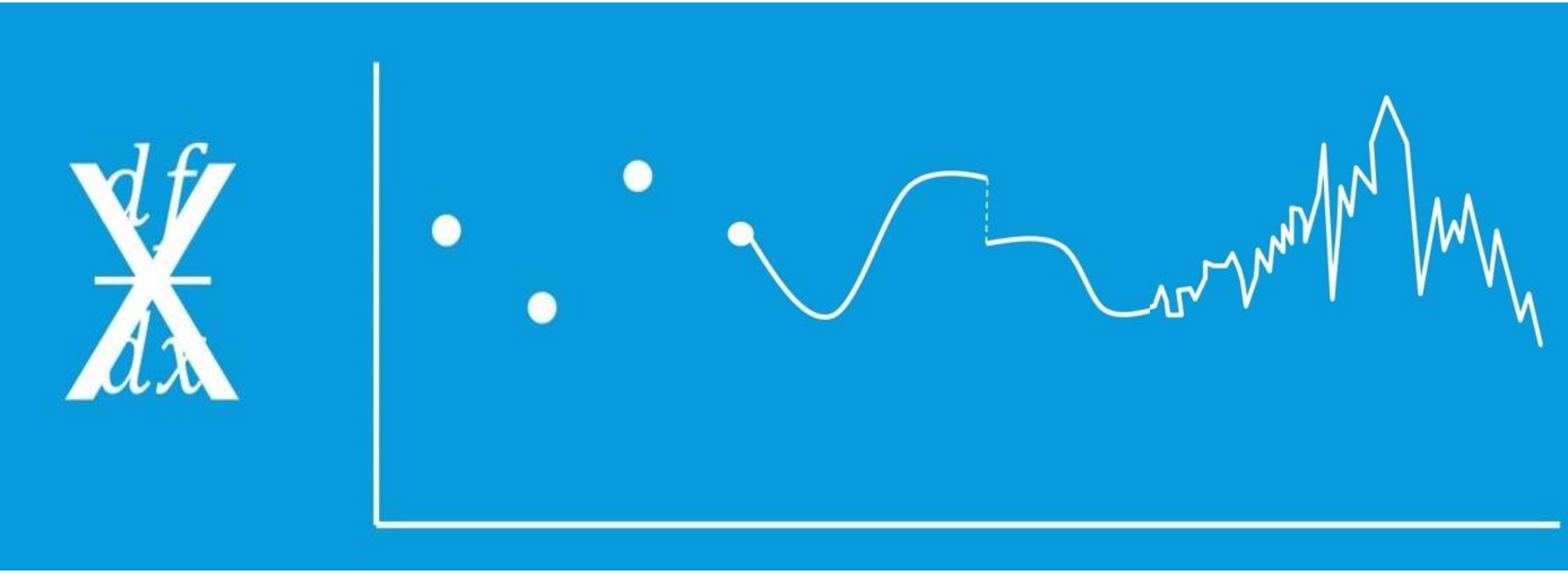
GRADIENT
FREE
ALGORITHMS

# GRADIENT FREE ALGORITHM

NO DERIVATIVES NEEDED

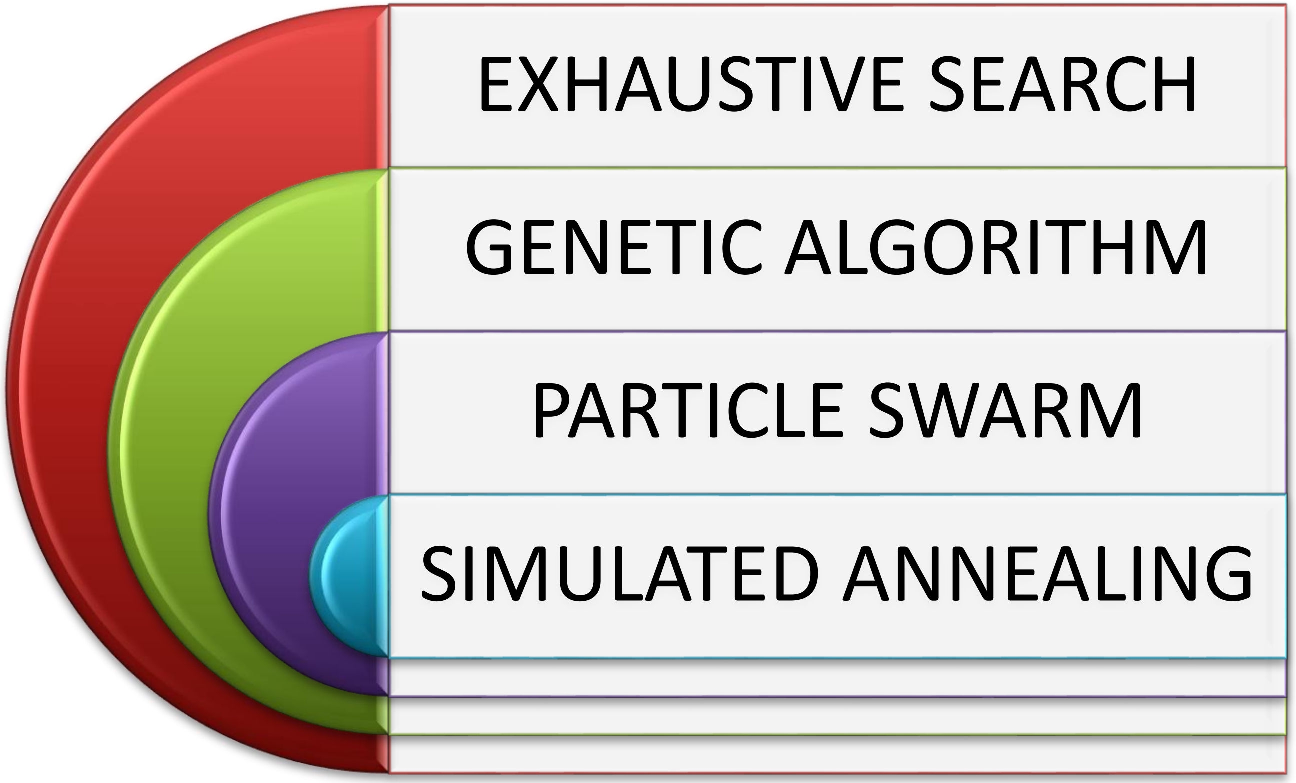
$$\frac{df^n}{dx} \quad or \quad f' \quad or \quad \dot{f}$$

# GRADIENT FREE ALGORITHM



# GRADIENT FREE ALGORITHM

# GRADIENT FREE ALGORITHM



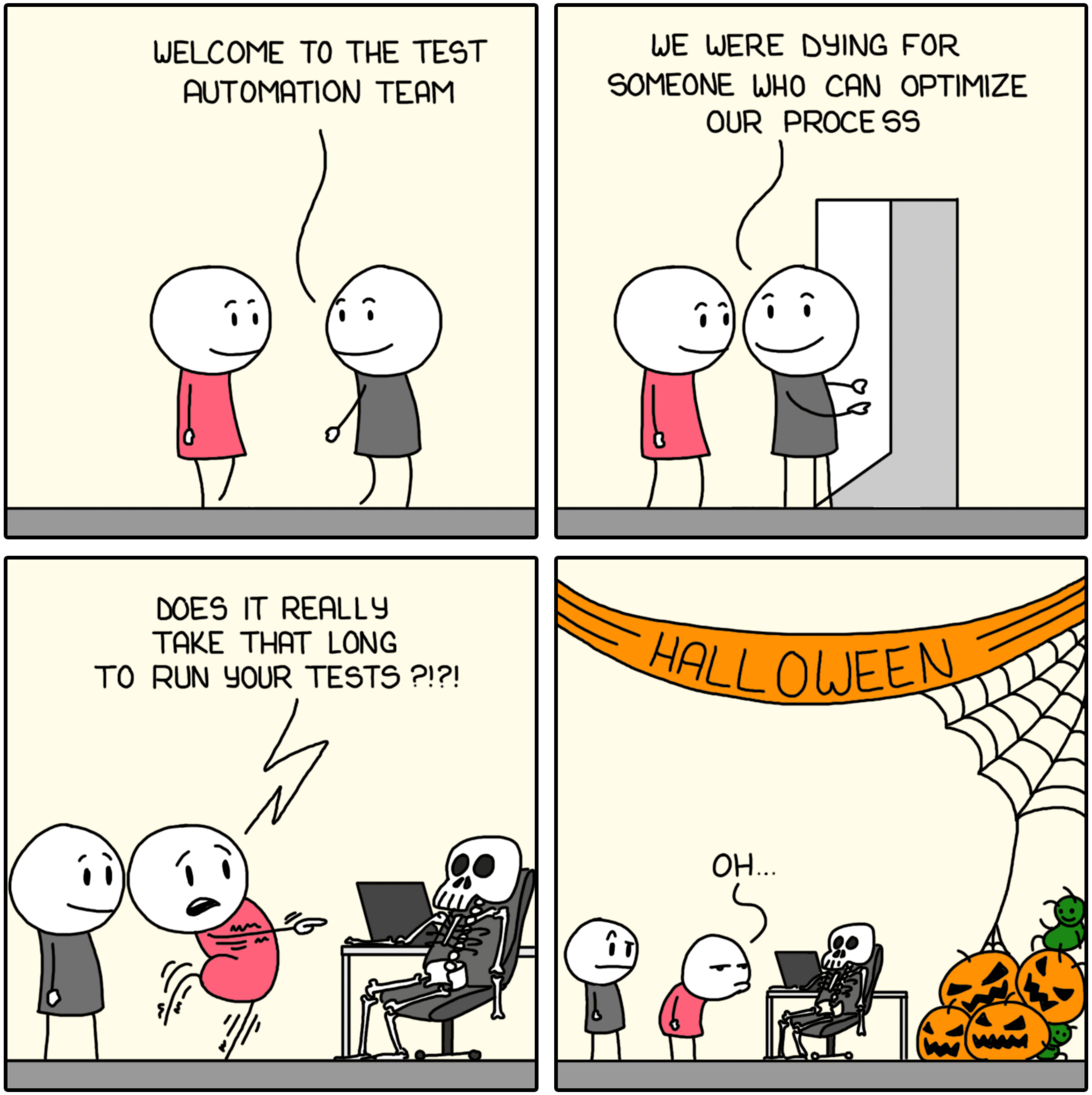
EXHAUSTIVE SEARCH

GENETIC ALGORITHM

PARTICLE SWARM

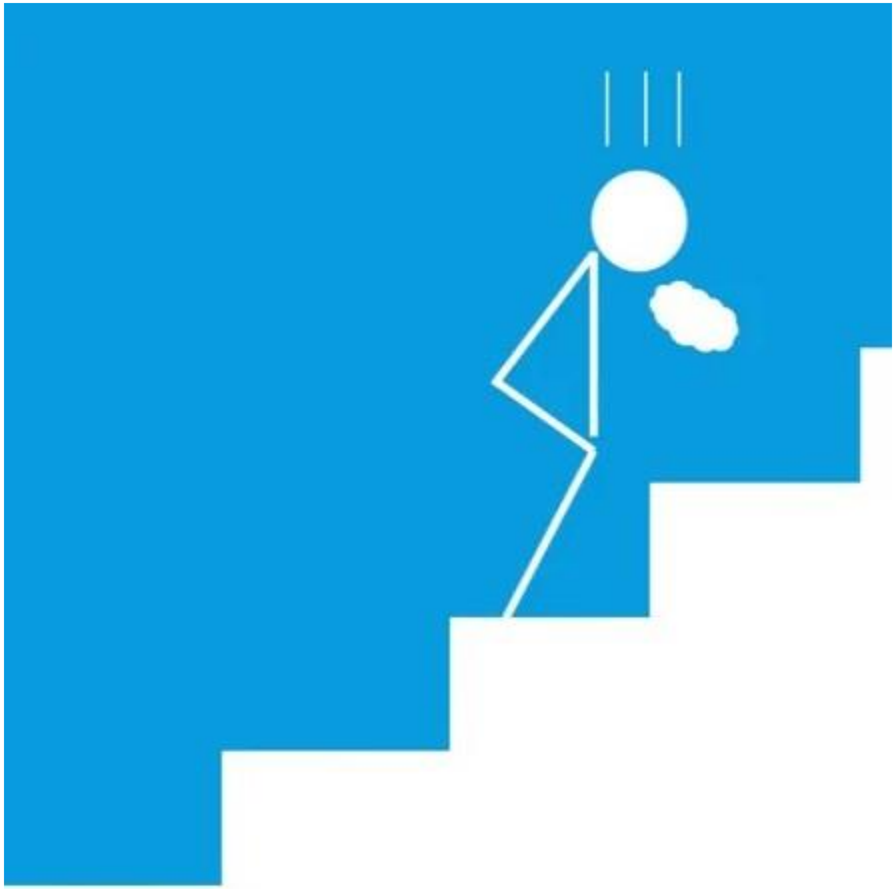
SIMULATED ANNEALING

# TESTS OPTIMIZATION

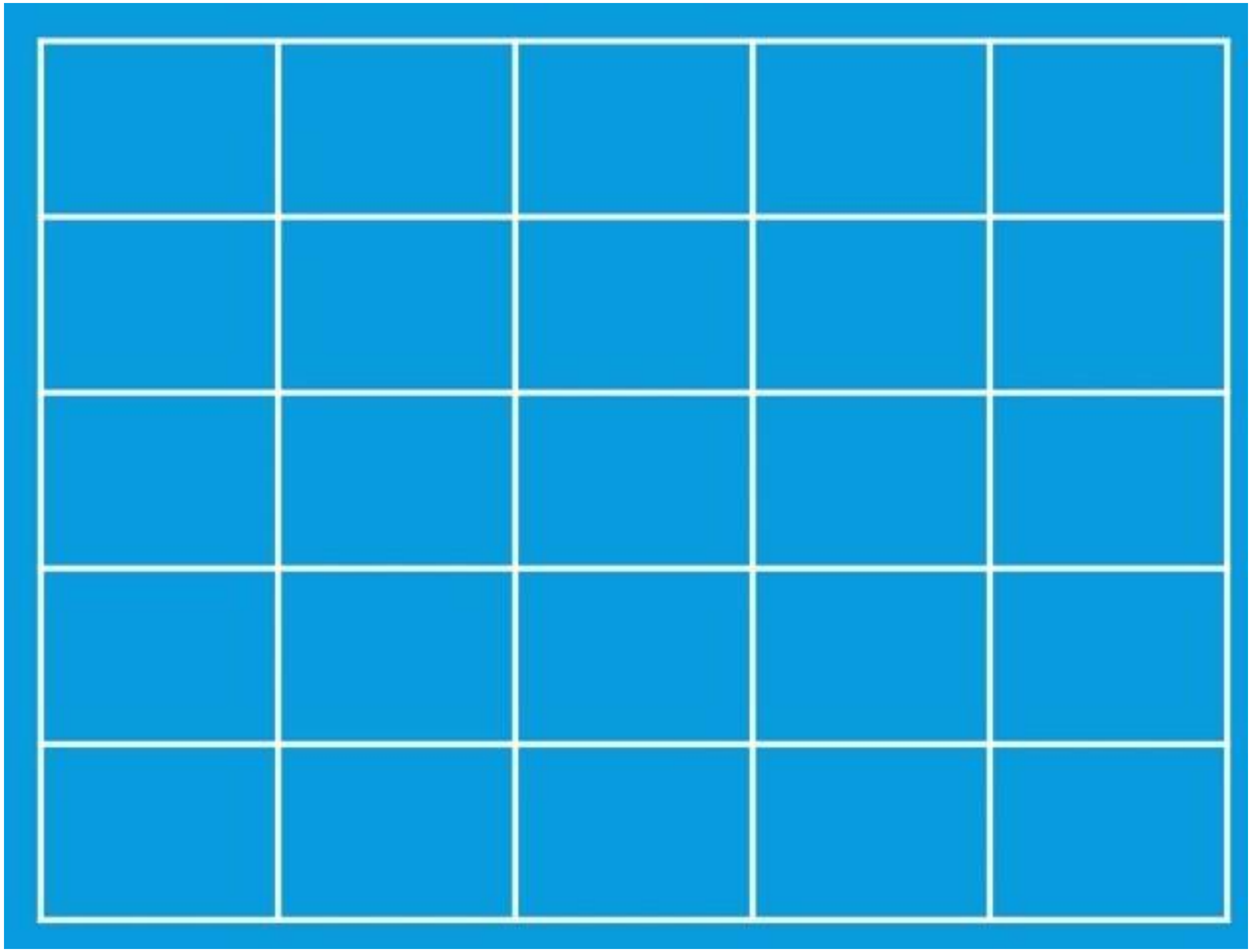

WELCOME TO THE TEST AUTOMATION TEAM
WE WERE DYING FOR SOMEONE WHO CAN OPTIMIZE OUR PROCESS
DOES IT REALLY TAKE THAT LONG TO RUN YOUR TESTS ?!?!
HALLOWEEN
OH...

MONKEYUSER.COM

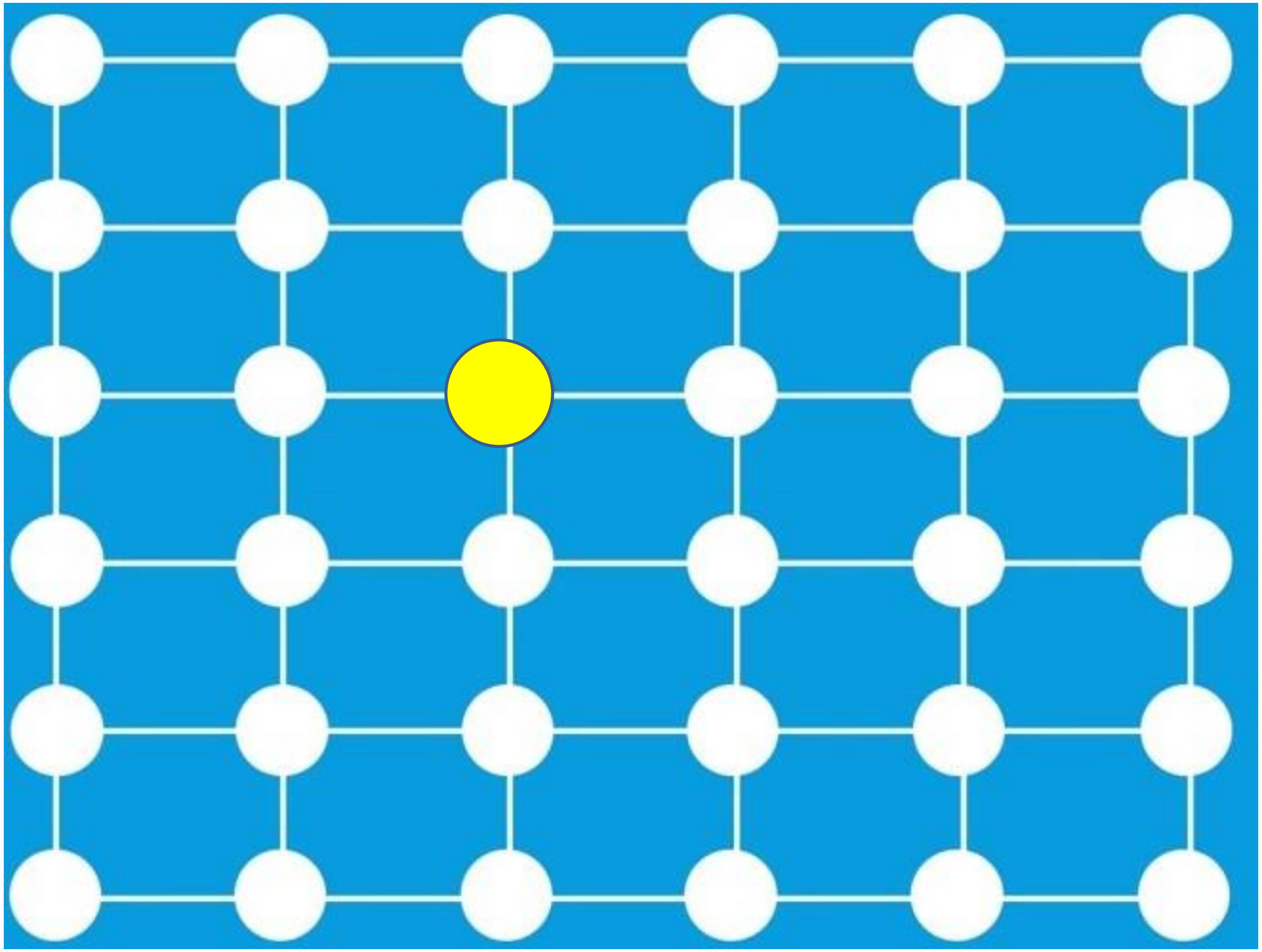
# EXHAUSTIVE SEARCH ALGORITHM



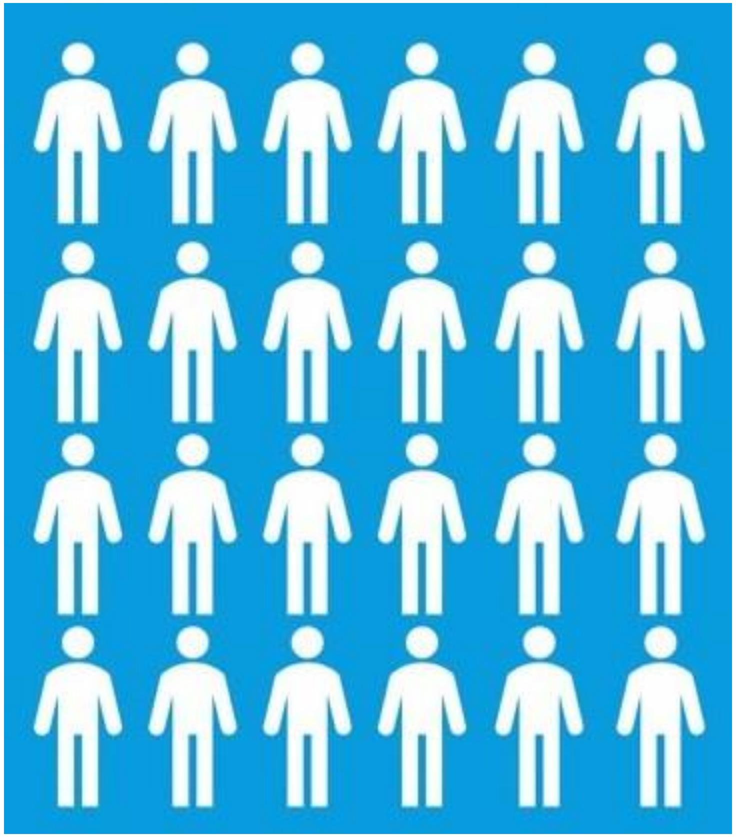
# EXHAUSTIVE SEARCH ALGORITHM



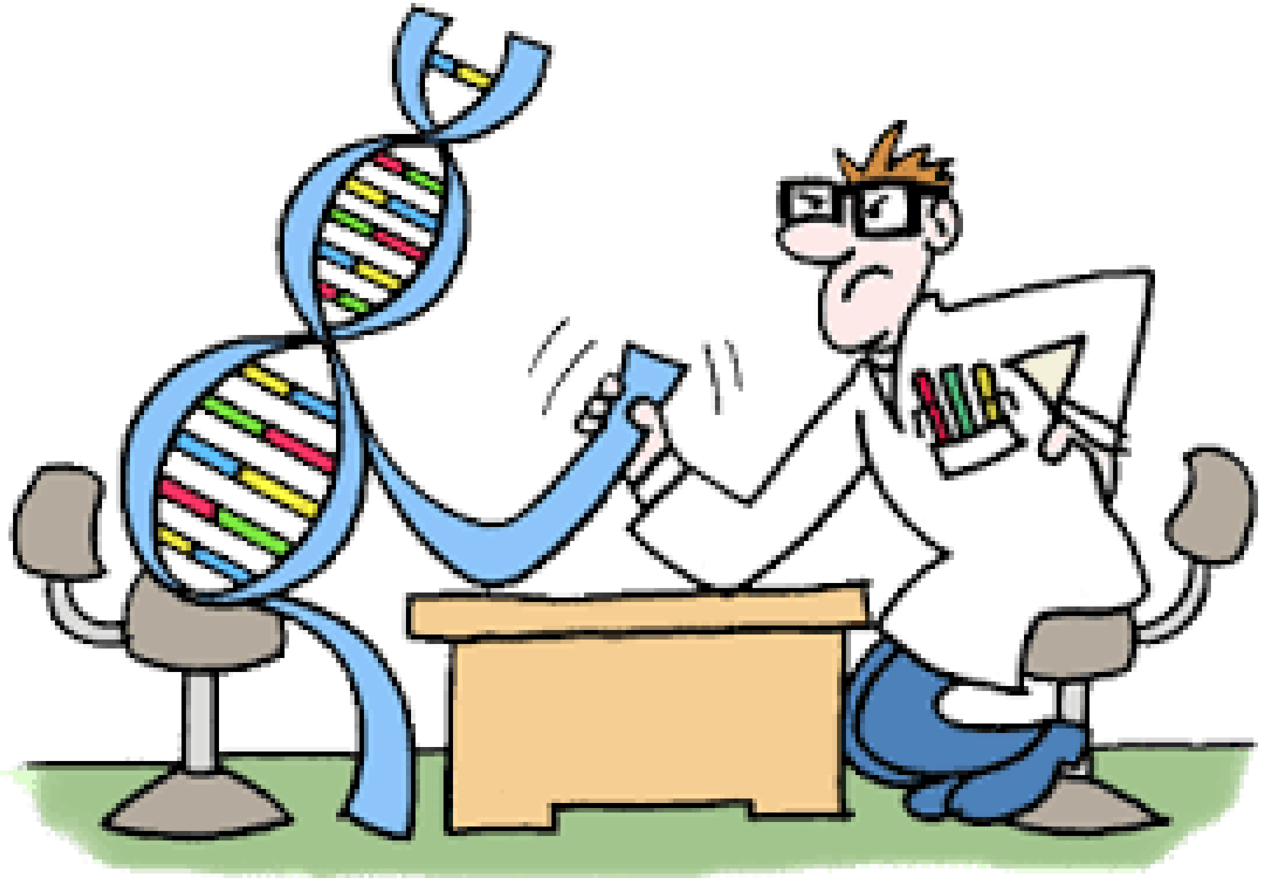
# EXHAUSTIVE SEARCH ALGORITHM



# GENETIC ALGORITHM



# GENETIC ALGORITHM

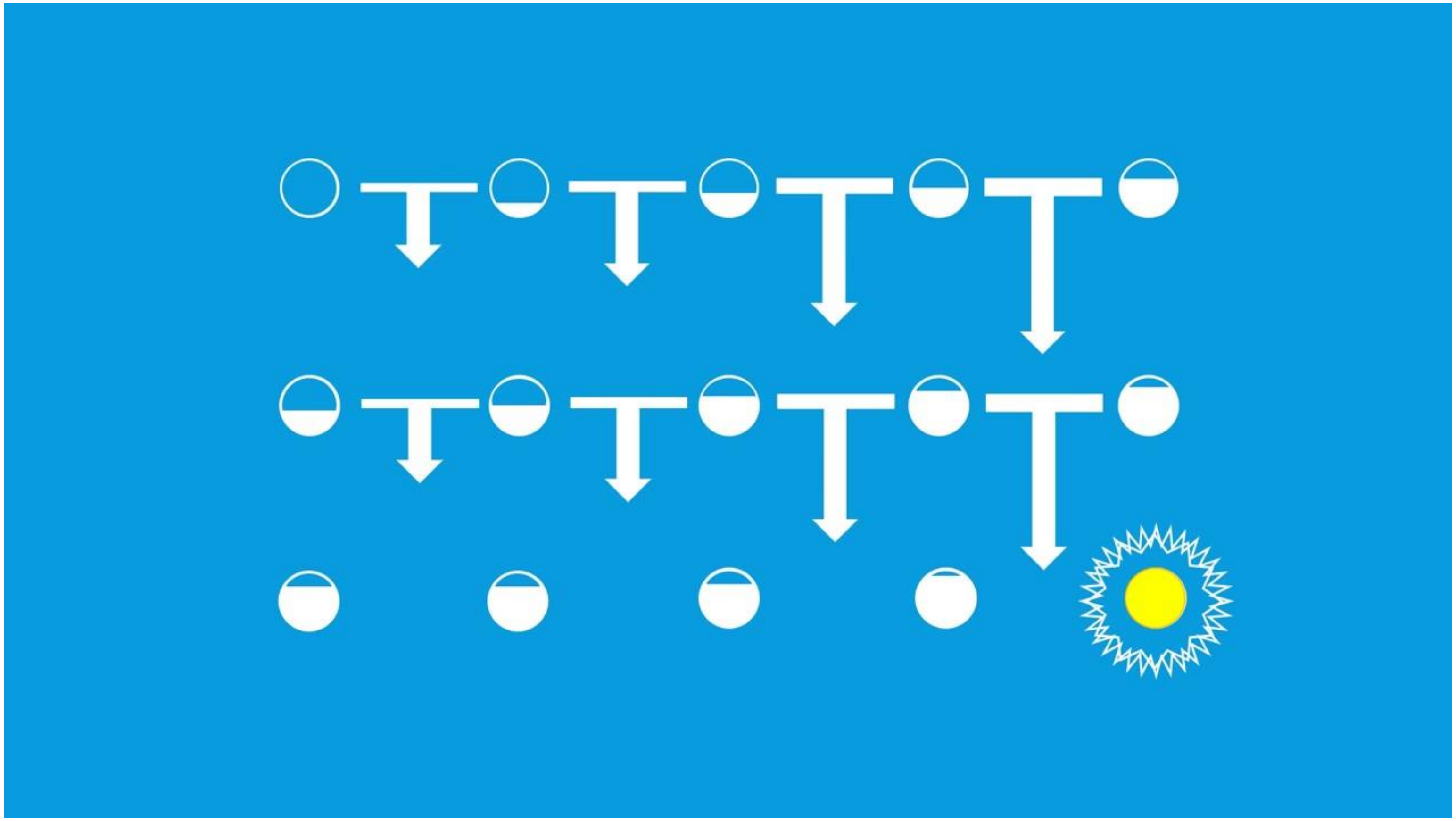


# WHY GA WORKS?

Mutation + Selection = Improvement!

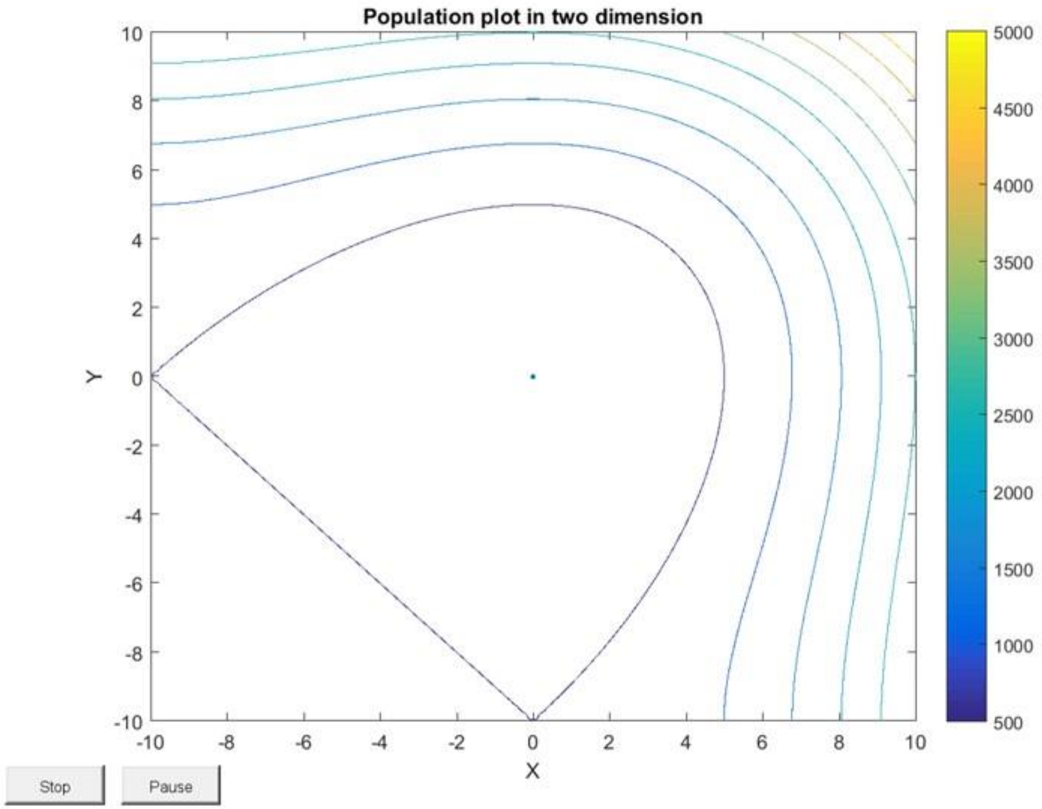
Crossover + Selection = Innovation!

# GENETIC ALGORITHM

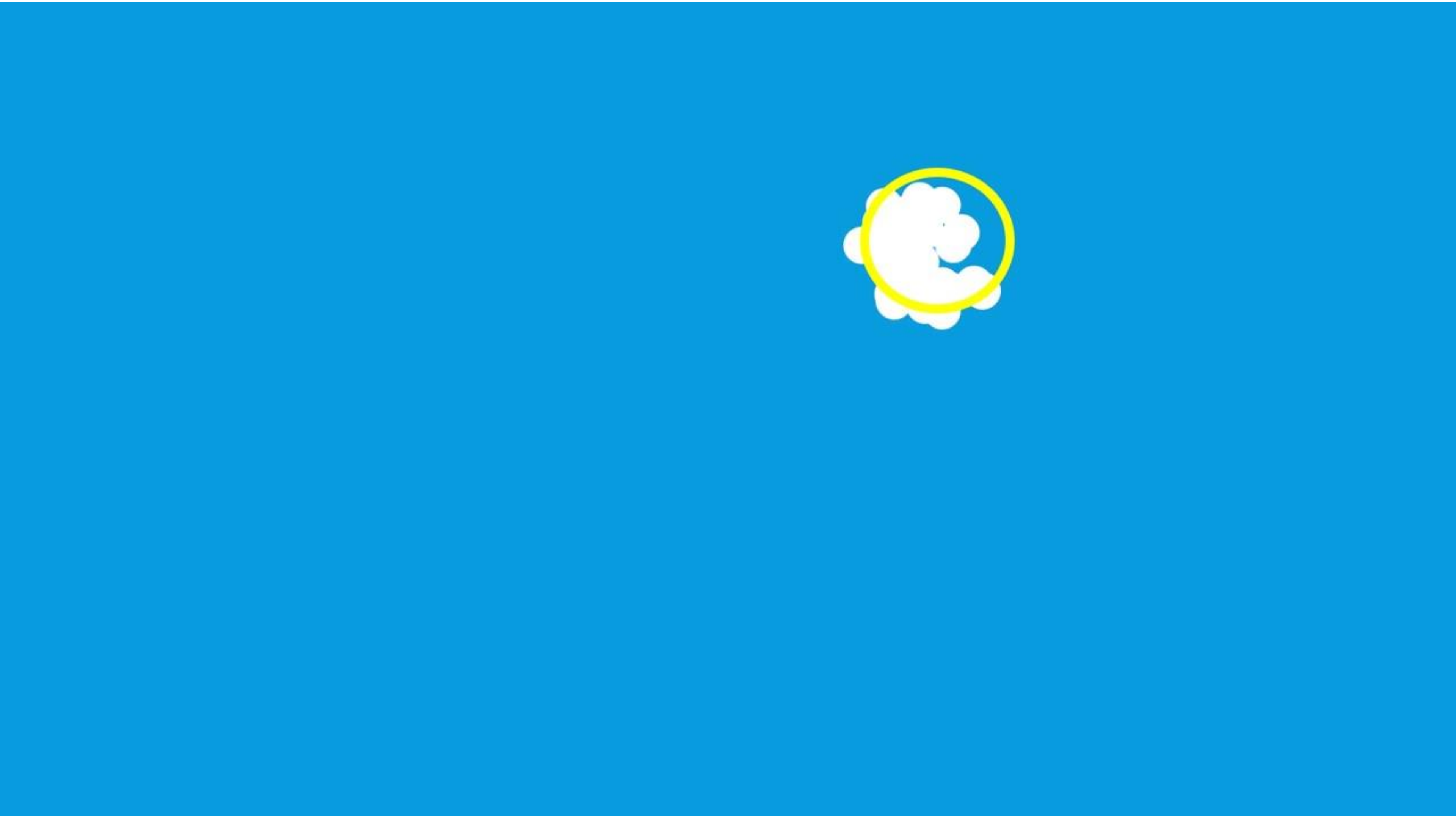


# GENETIC ALGORITHM

$$f(x,y) = x^3 + 15x^2 + y^3 + 15y^2$$



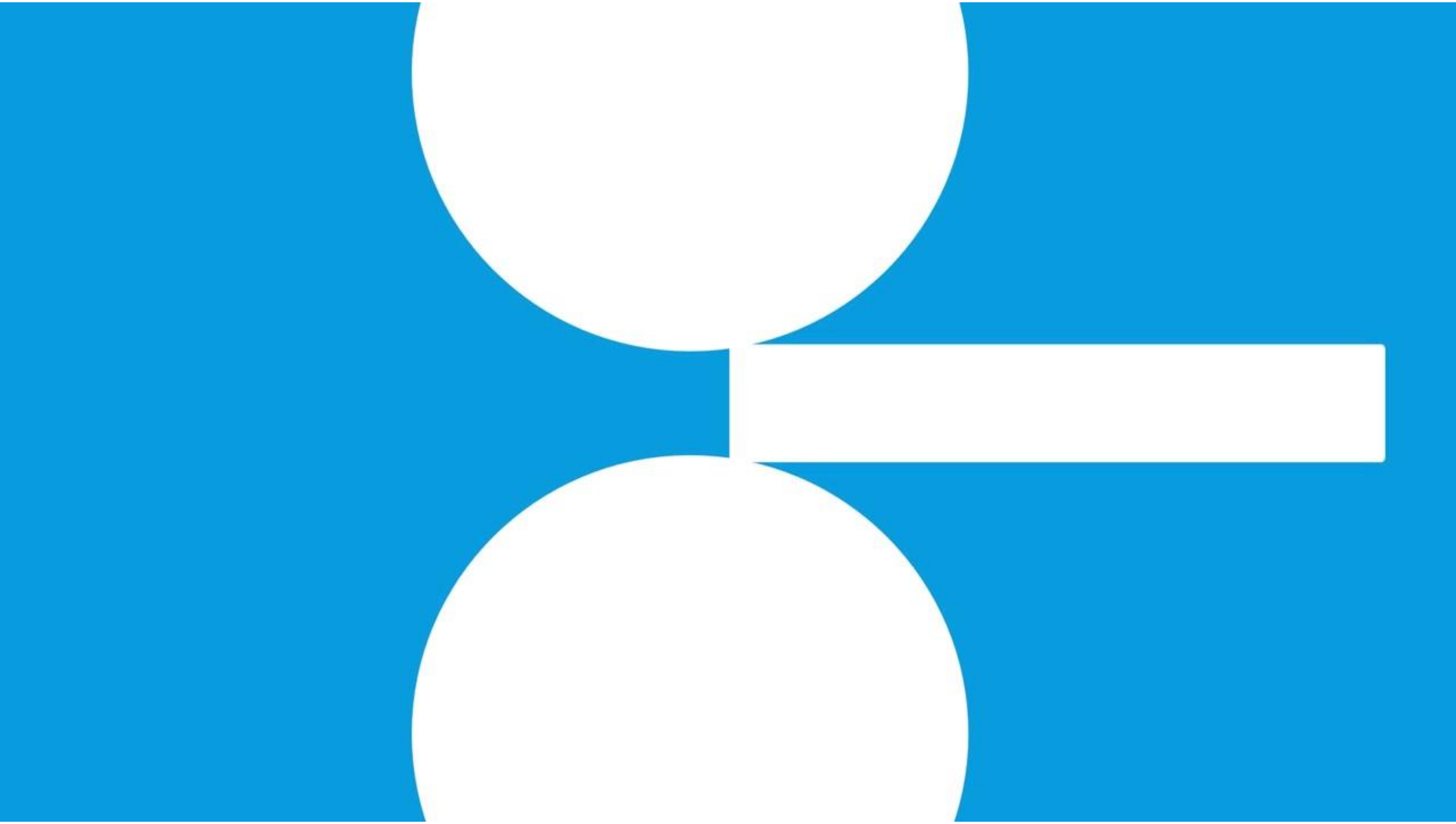
# PARTICLE SWARM



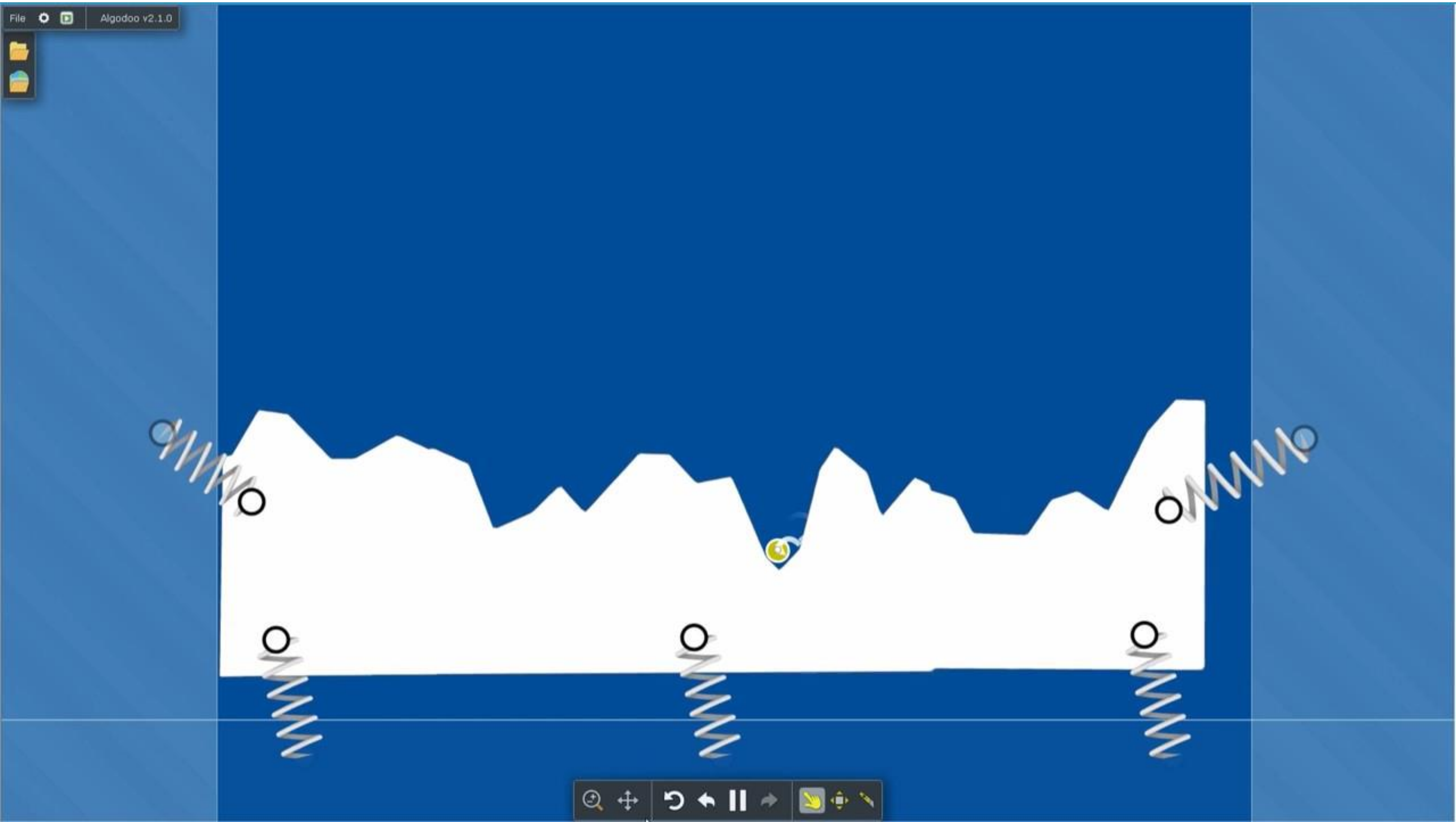
# GRADIENT FREE ALGORITHM

SIMULATED ANNEALING

# SIMULATED ANNEALING



# SIMULATED ANNEALING



# OTHER GRADIENT FREE ALGORITHM BASED ON NATURAL PROCESSES

- ANT COLONY OPTIMIZATION
- PARTICLE SWARM
- HARMONY SEARCH
- ARTIFICIAL BEE COLONY
- BEES ALGORITHM
- SHUFFLED FROG
- IMPERIALISTIC COMPETITIVE
- RIVER FORMATION DYNAMICS
- INTELLIGENT WATER DROPS ALGO.
- GRAVITATIONAL SEARCH ALGO.
- BAT ALGO.
- FLOWER FORMATION
- CUTTLE FISH ALGO.

HAPPINESS IS ASSUMING THE WORLD IS LINEAR

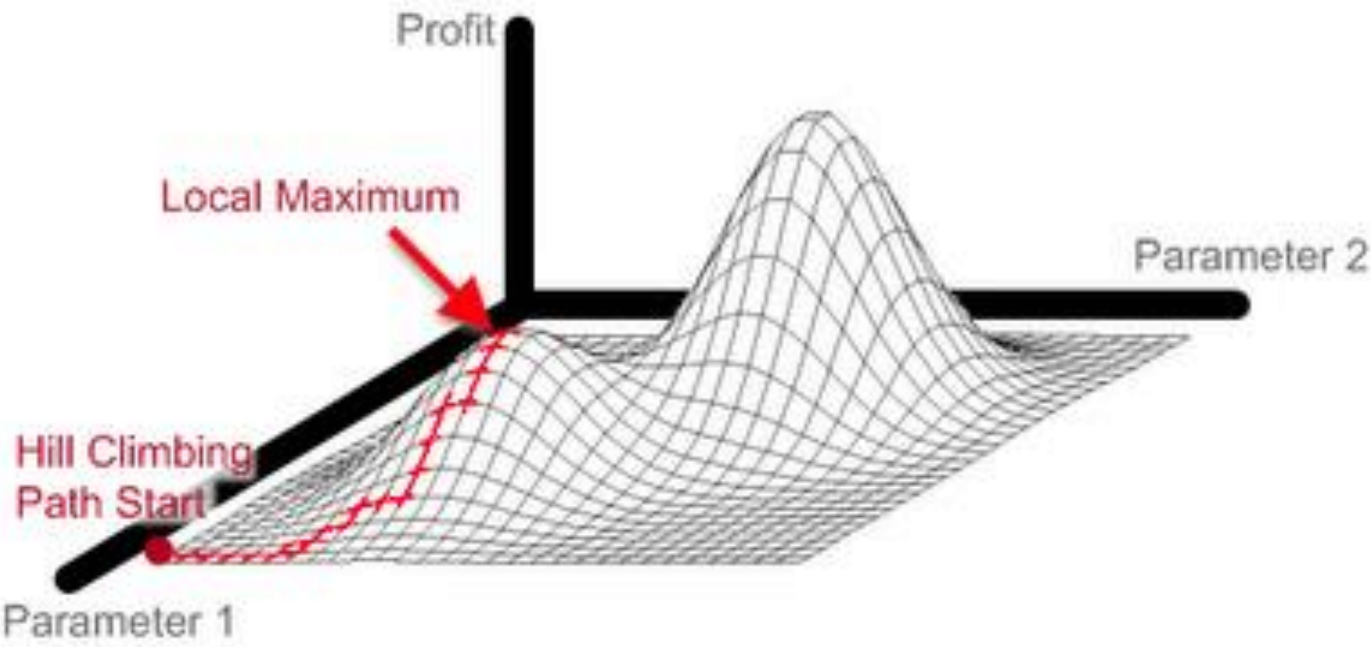
# Optimality criteria

a) **Local Optimal Point:** A point/solution or solution $x^*$ is said to be local optimal point if there exist no point in the neighbourhood of $x^*$ which is better than $x^*$.

b) **Global Optimal Point:** If there exists no point in the entire search space which s better than $x^{**}$.

# A "local" maximum [in contrast to a "global" maximum] is a point that's higher than what's around it but not actually the highest.

The problem with hill climbing is that it gets stuck on "local-maxima"



Source:www.maxdama.com/2008/07/trading-optimization

# Improving Results and Optimization

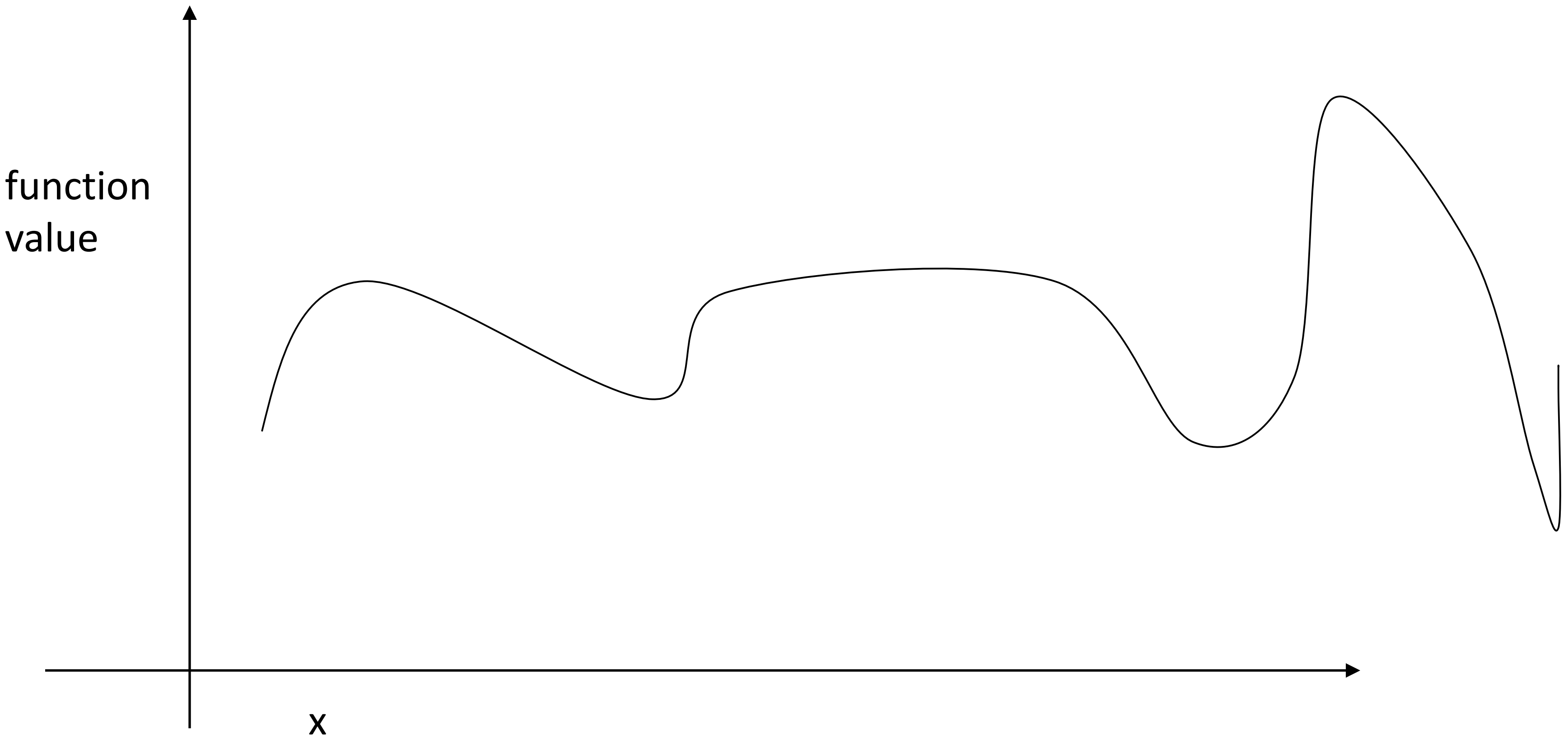
- Assume a state with many variables
- Assume some function that you want to maximize/minimize the value of
- Searching entire space is too complicated
  - Can’t evaluate every possible combination of variables
  - Function might be difficult to evaluate analytically

# 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

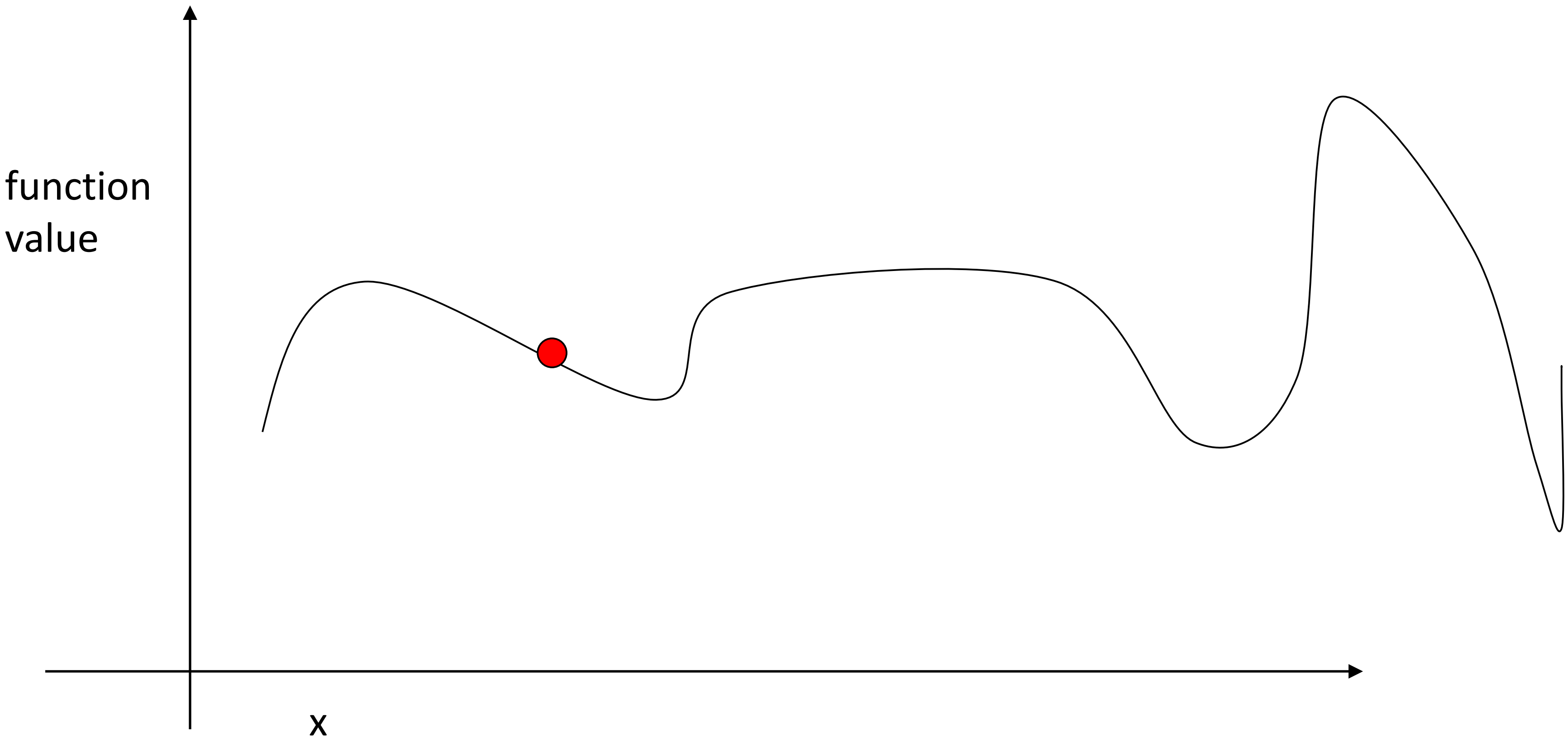
# Simple Example

- One dimension (typically use more):



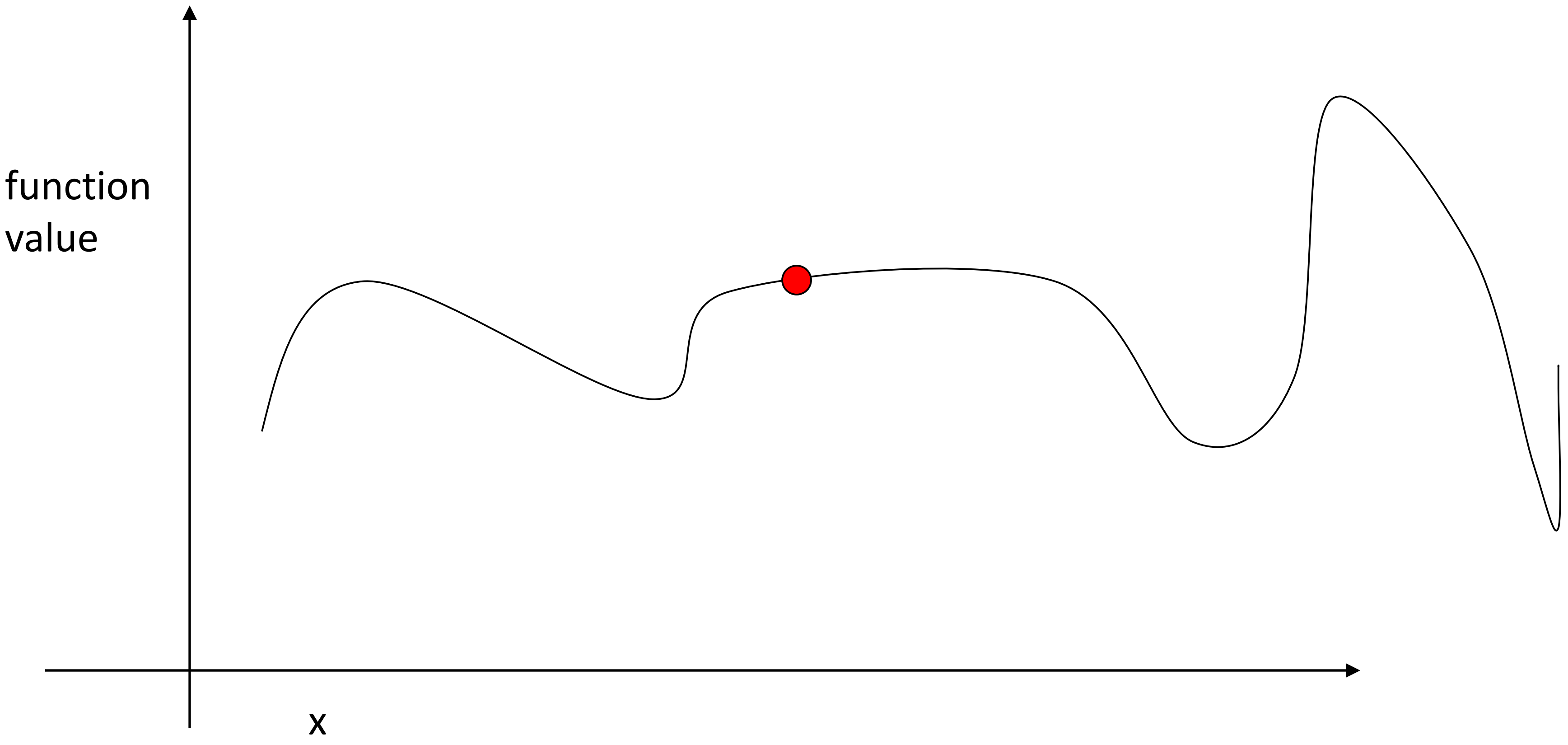
# Simple Example

- Start at a valid state, try to maximize



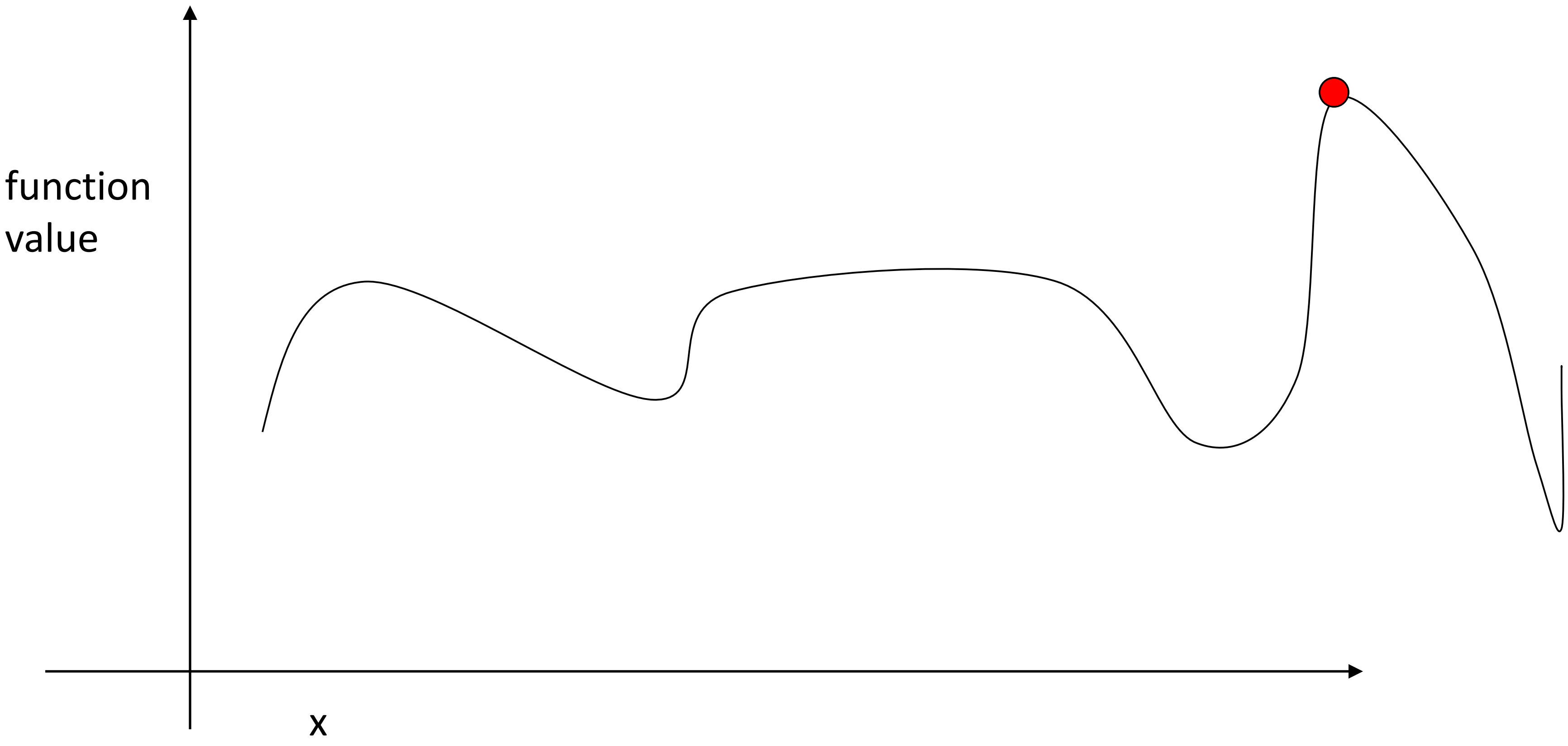
# Simple Example

- Move to better state



# Simple Example

- Try to find maximum

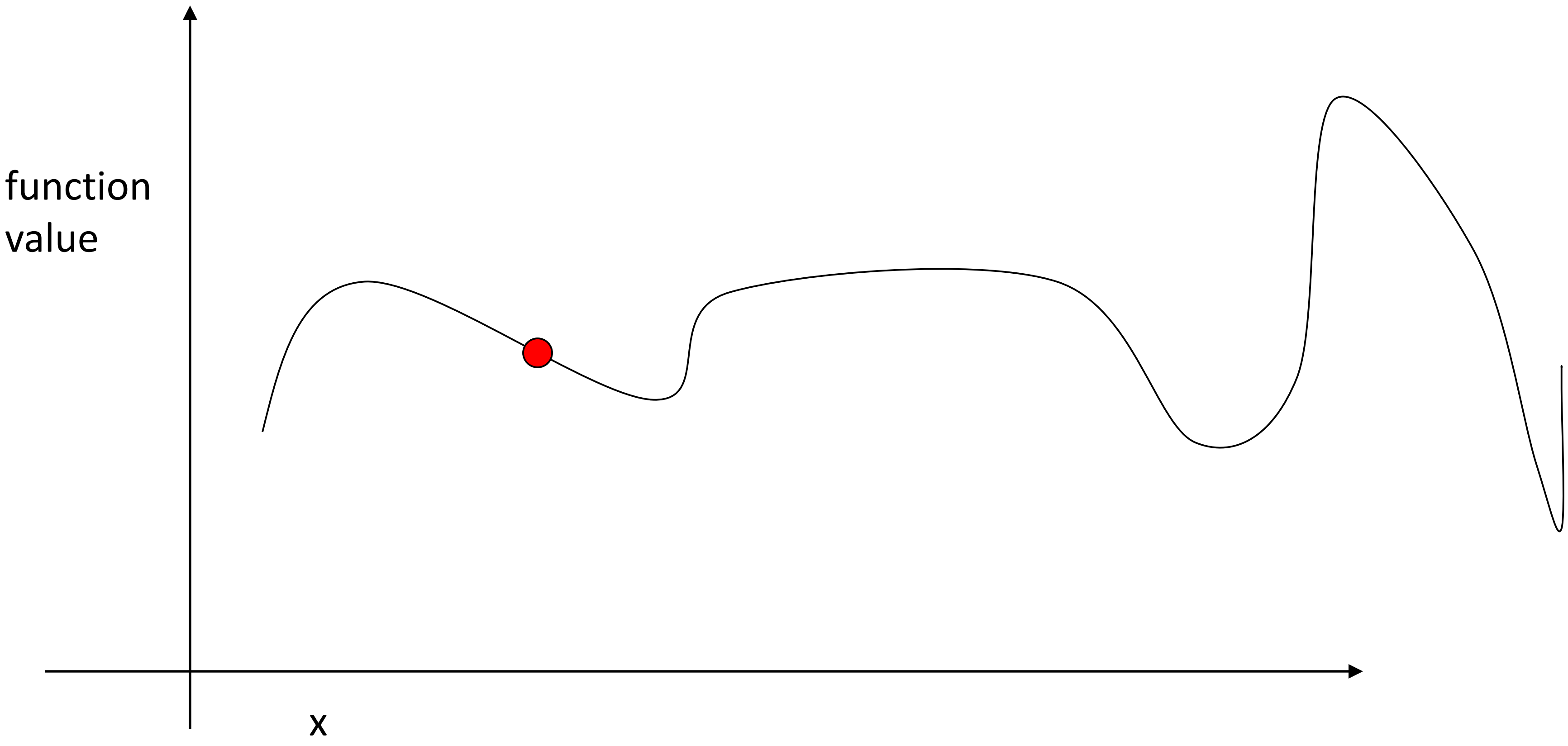


# Hill-Climbing

```
Choose Random Starting State
Repeat
    From current state, generate n random steps in random directions
    Choose the one that gives the best new value
While some new better state found
(i.e. exit if none of the n steps were better)
```

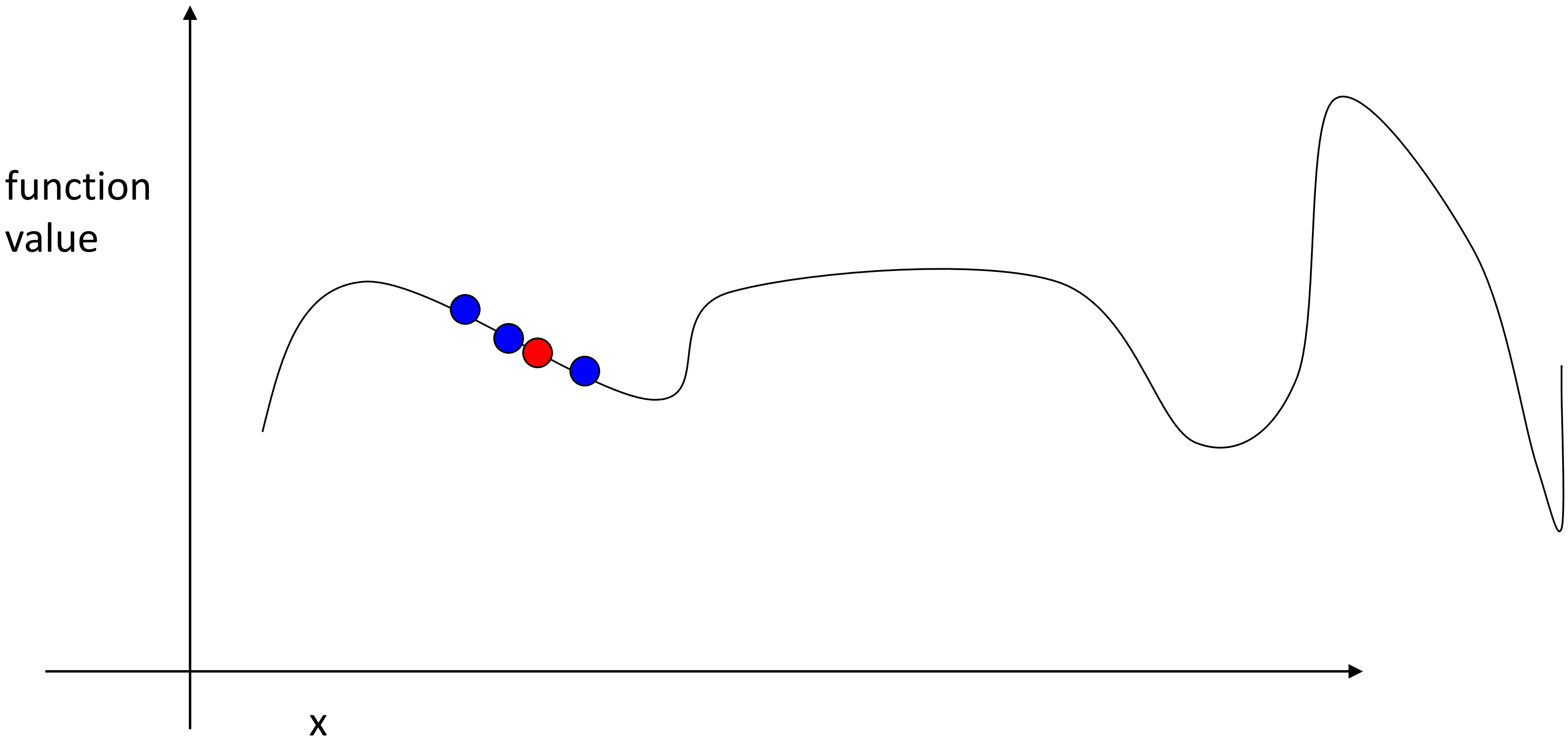
# Simple Example

- Random Starting Point



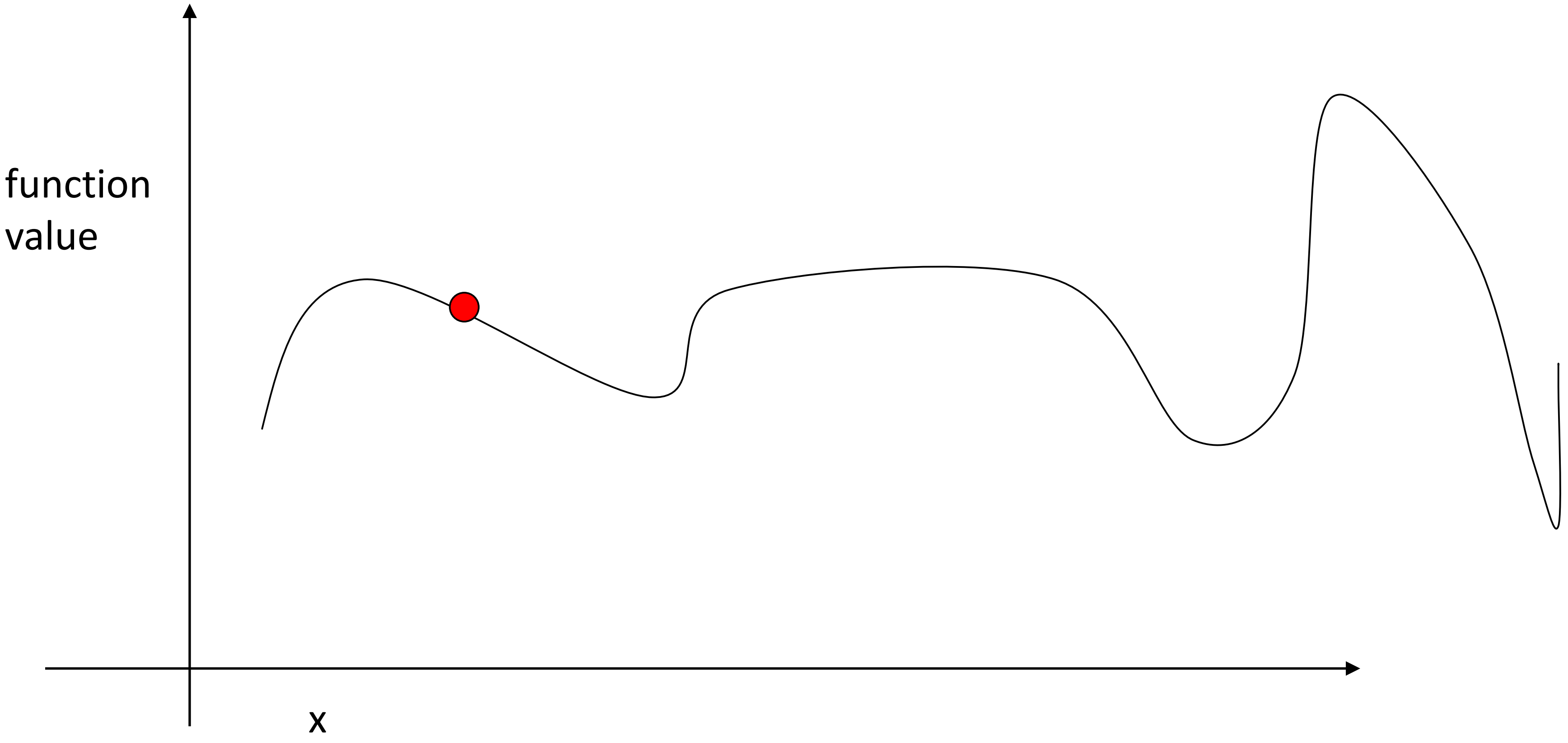
# Simple Example

- Three random steps



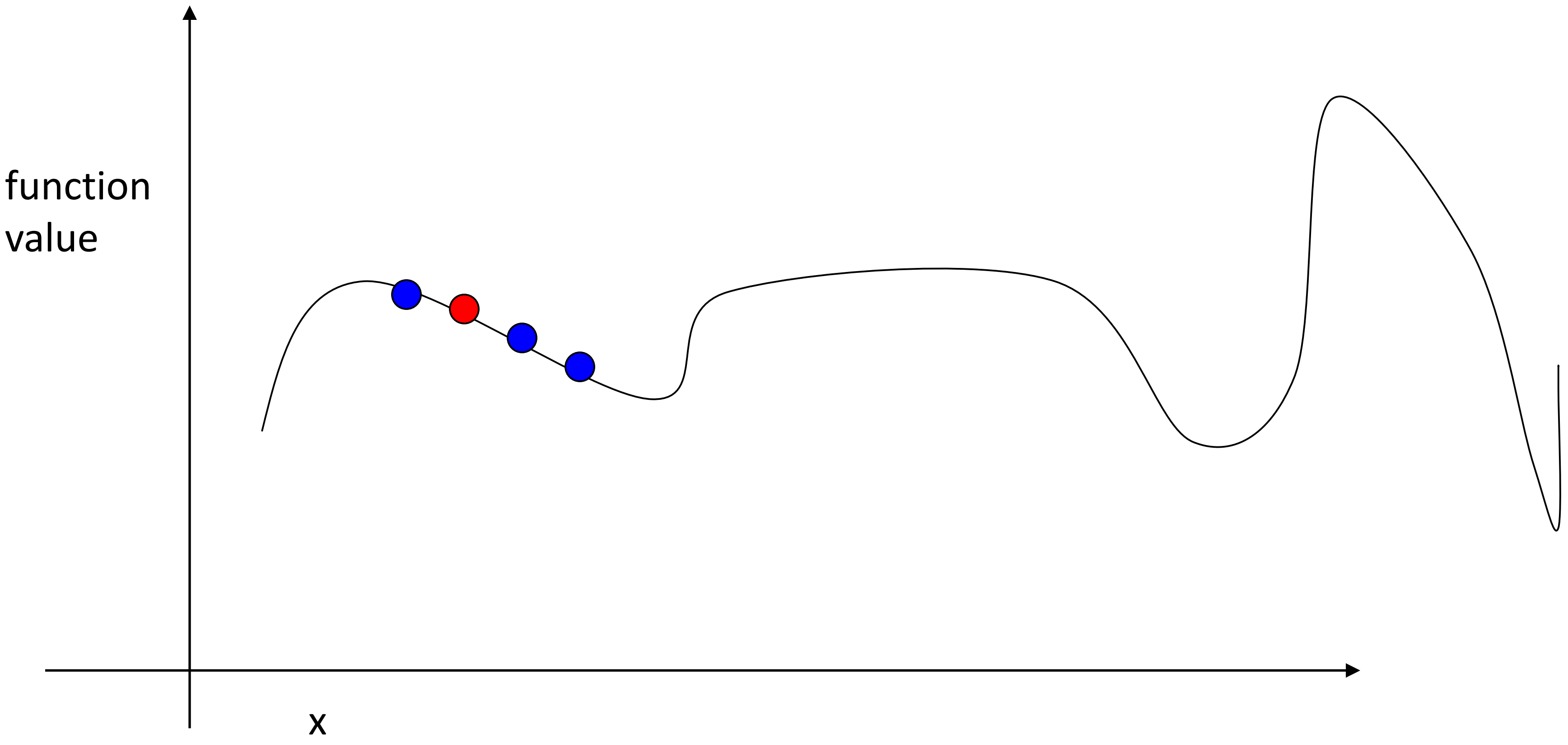
# Simple Example

- Choose Best One for new position



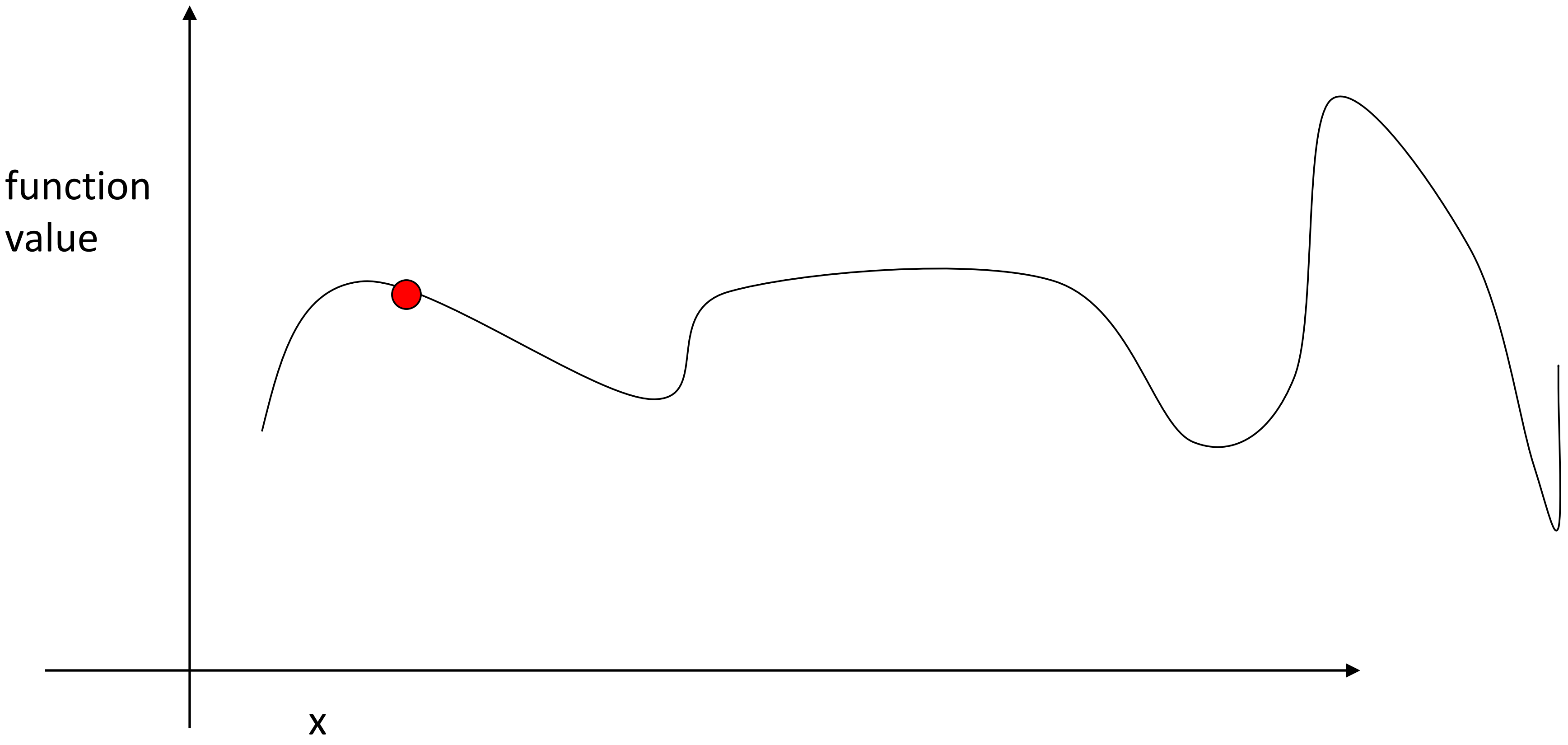
# Simple Example

- Repeat



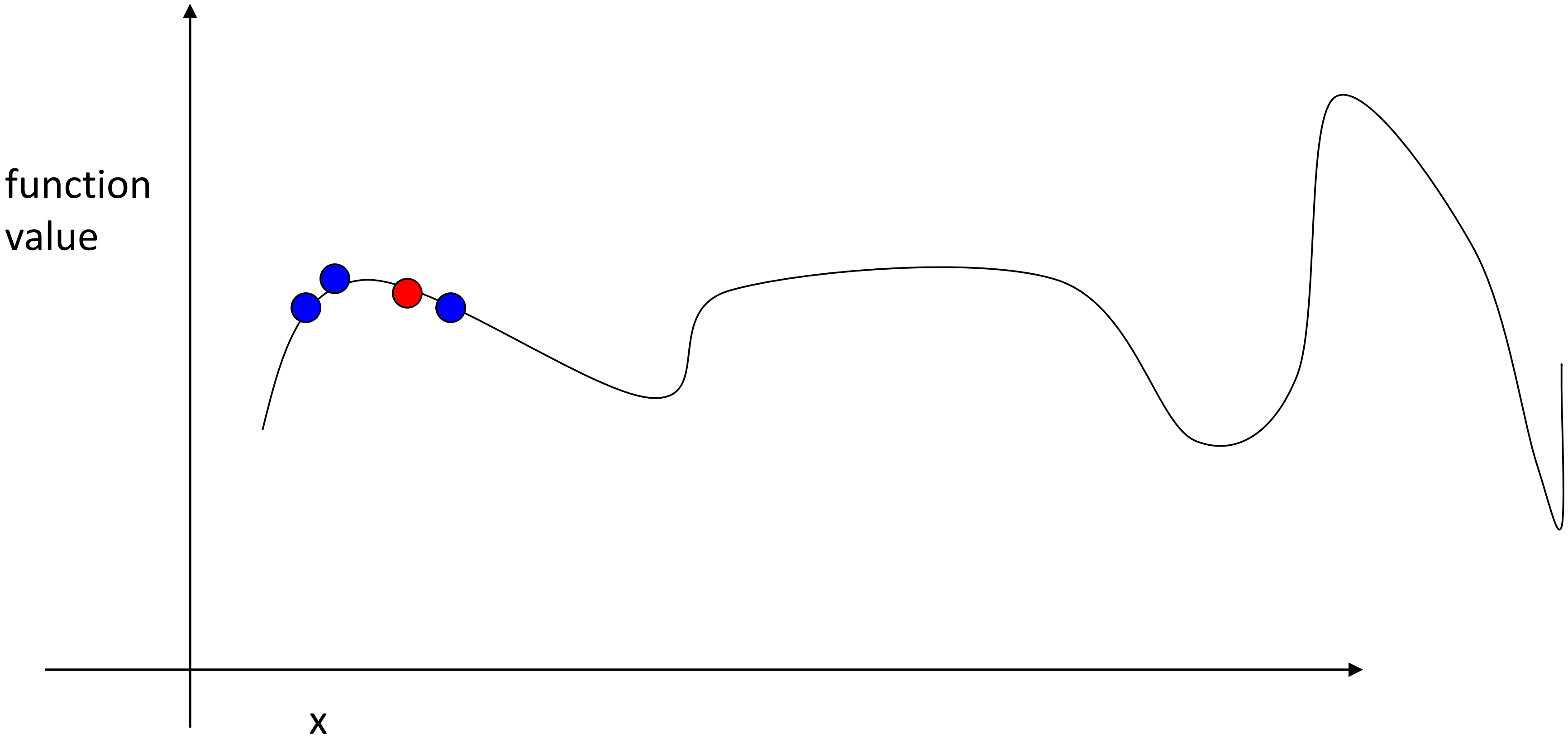
# Simple Example

- Repeat



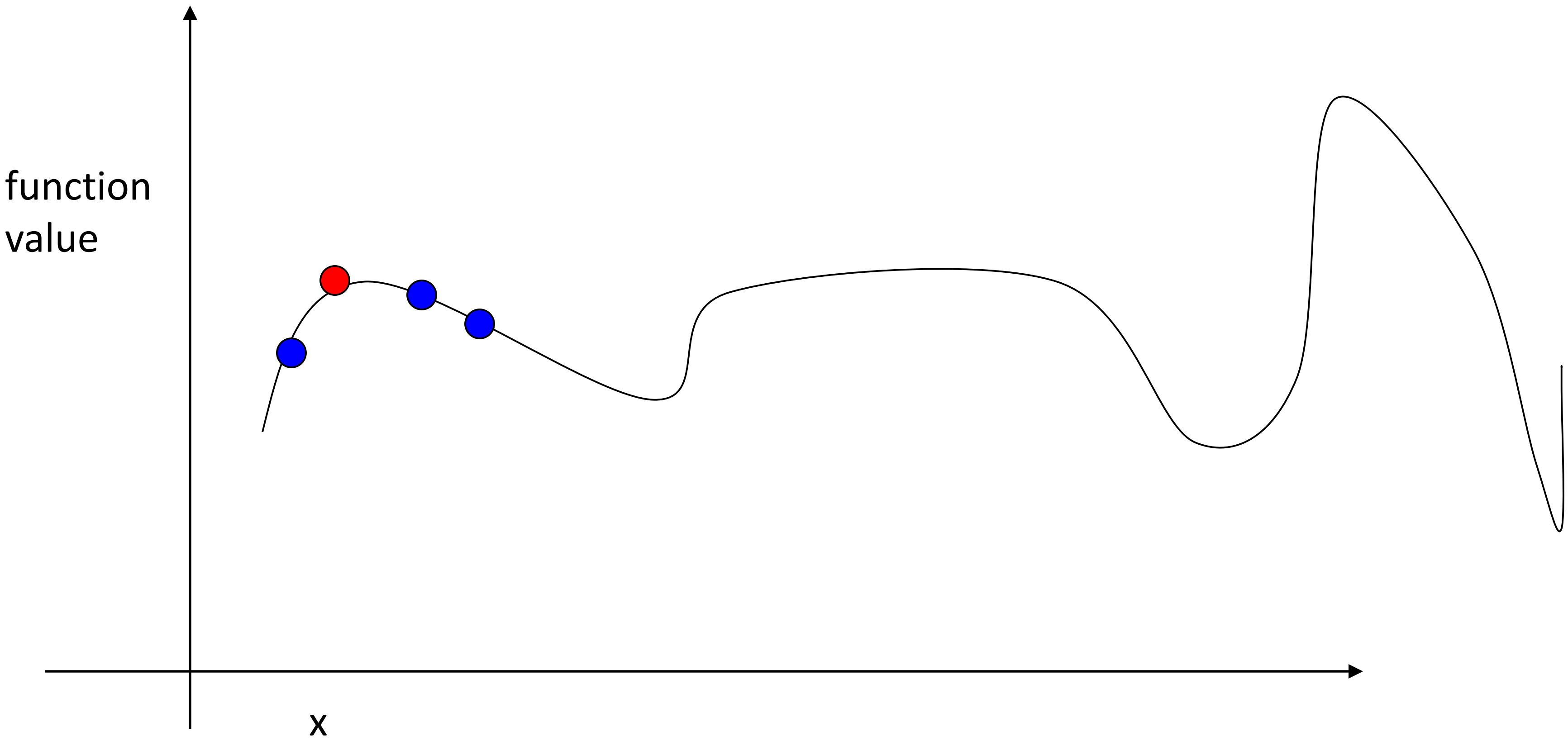
# Simple Example

- Repeat



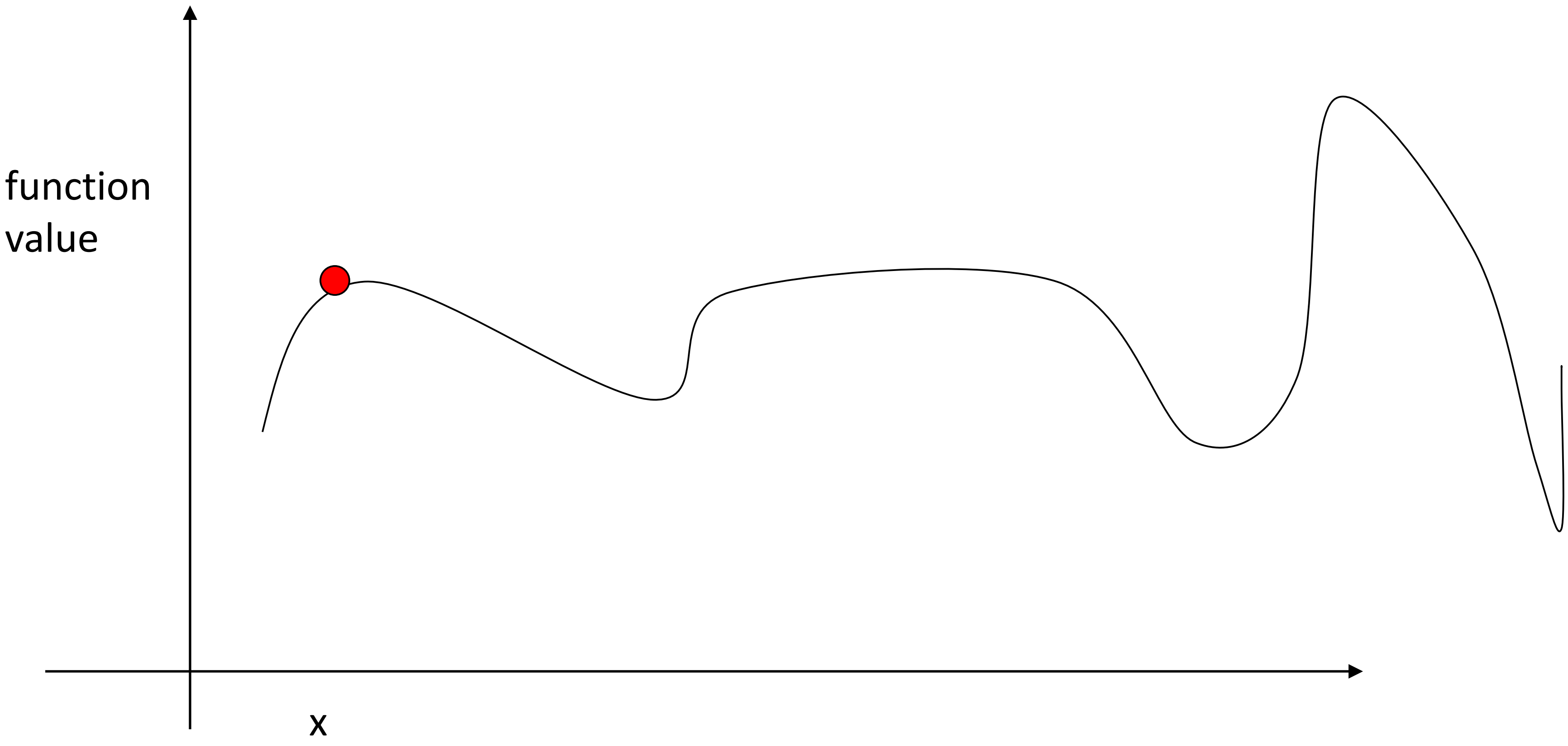
# Simple Example

- Repeat



# Simple Example

- No Improvement, so stop.



# Problems With Hill Climbing

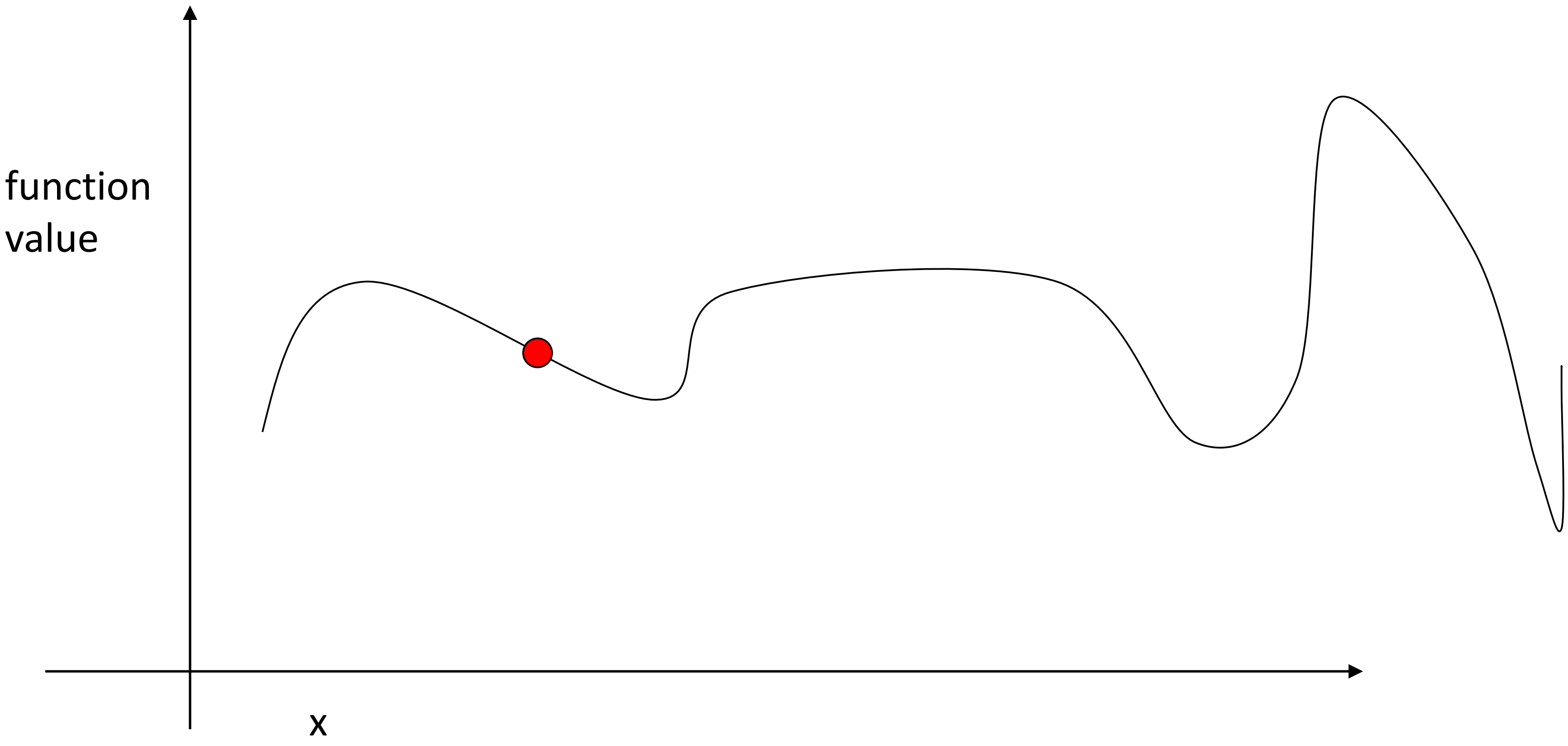
- Random Steps are Wasteful
  - Addressed by other methods
- Local maxima, plateaus, ridges
  - Can try random restart locations
  - Can keep the *n* best choices (this is also called "beam search")
- Comparing to game trees:
  - Basically looks at some number of available next moves and chooses the one that looks the best at the moment
  - Beam search: follow only the best-looking *n* moves

# Gradient Descent (or Ascent)

- Simple modification to Hill Climbing
  - Generallly assumes a continuous state space
- Idea is to take more intelligent steps
- Look at local gradient: the direction of largest change
- Take step in that direction
  - Step size should be proportional to gradient
- Tends to yield much faster convergence to maximum

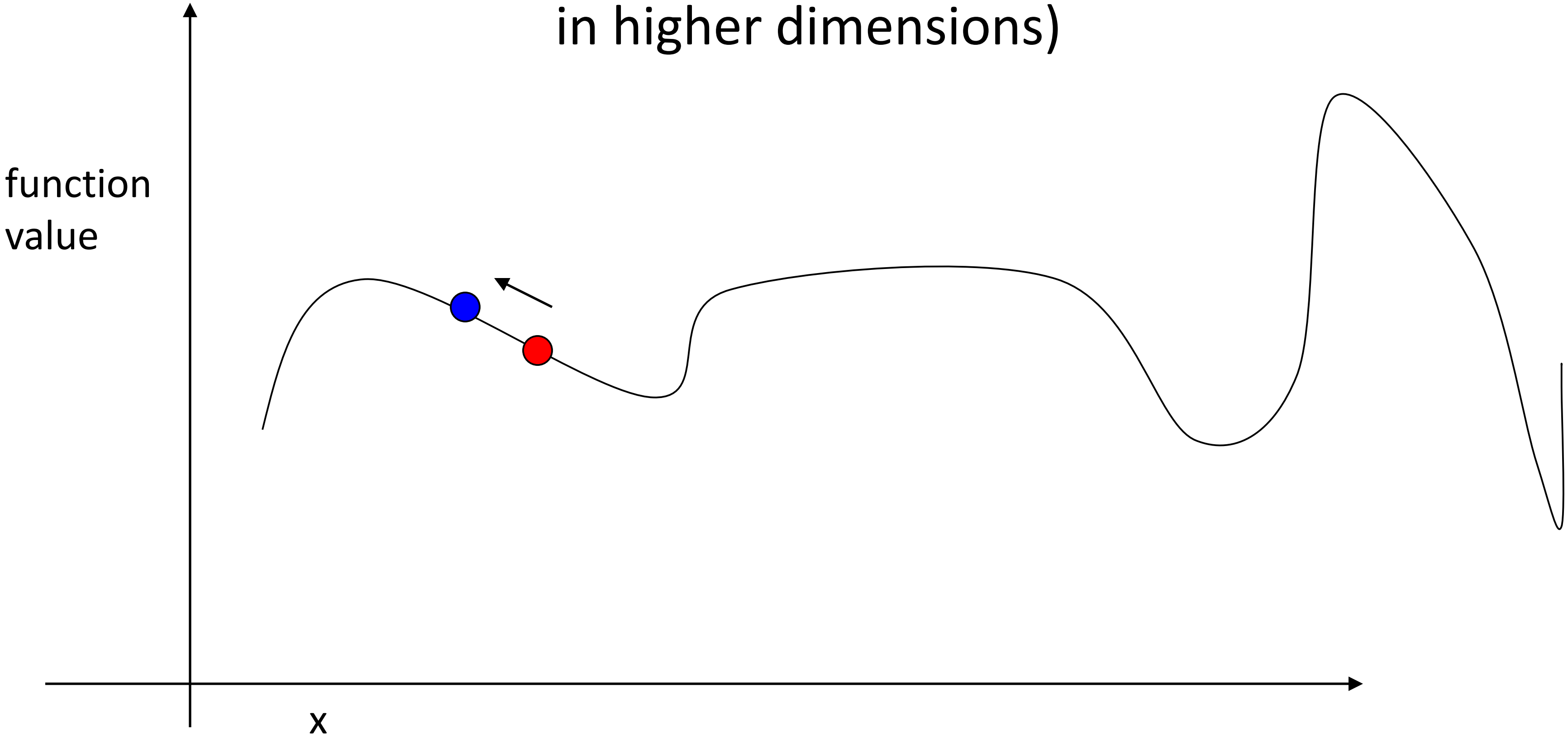
# Gradient Ascent

- Random Starting Point



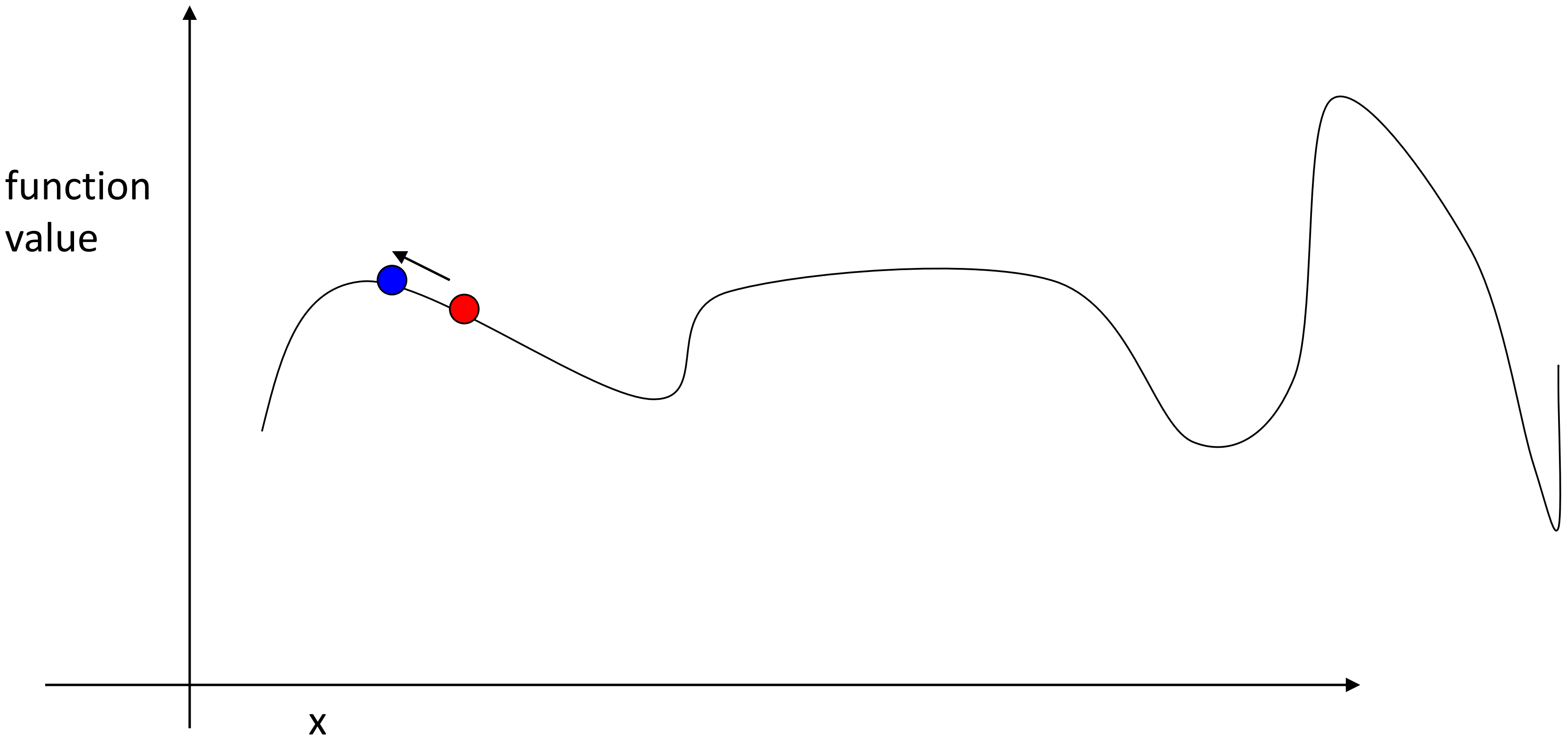
# Gradient Ascent

- Take step in direction of largest increase
  (obvious in 1D, must be computed in higher dimensions)



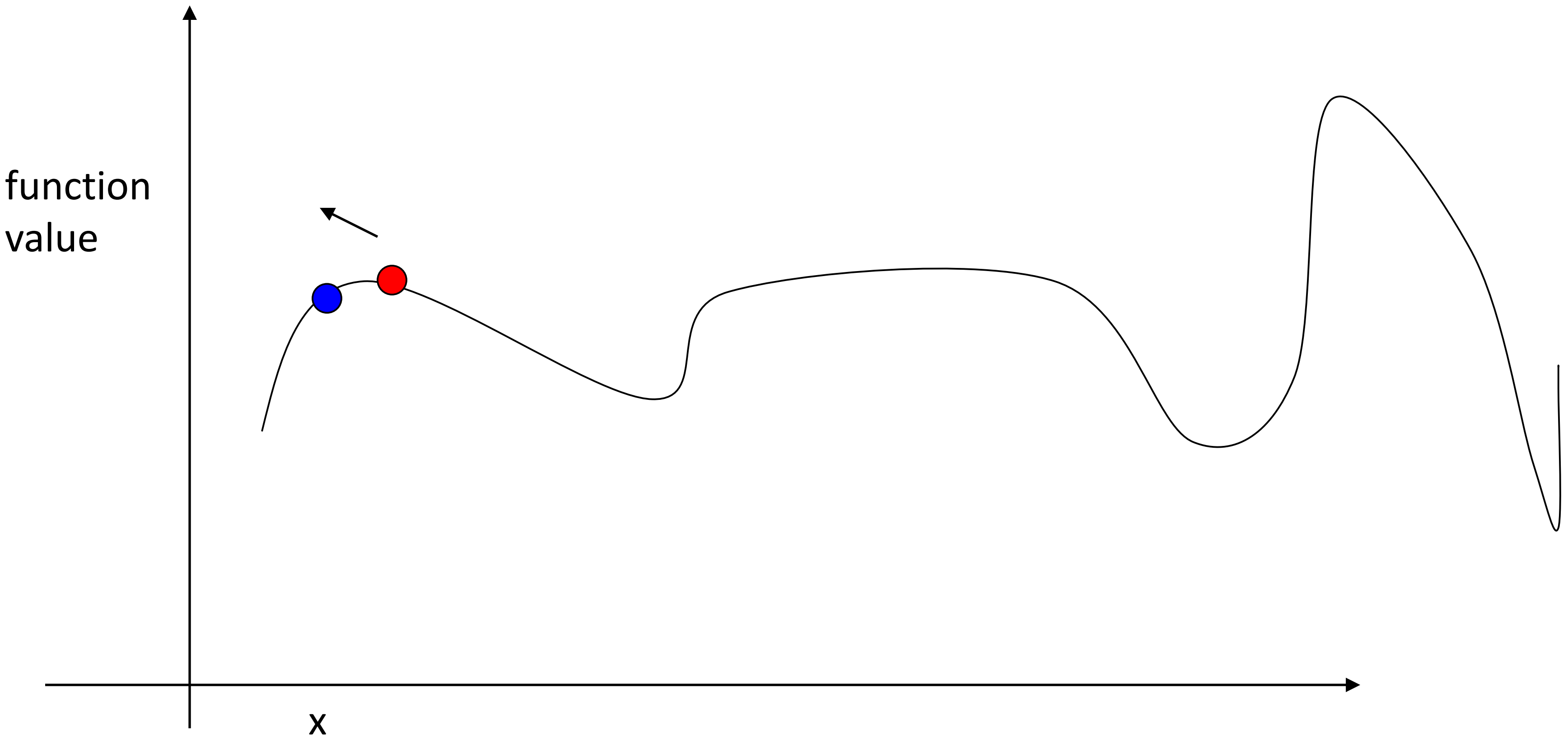
# Gradient Ascent

- Repeat



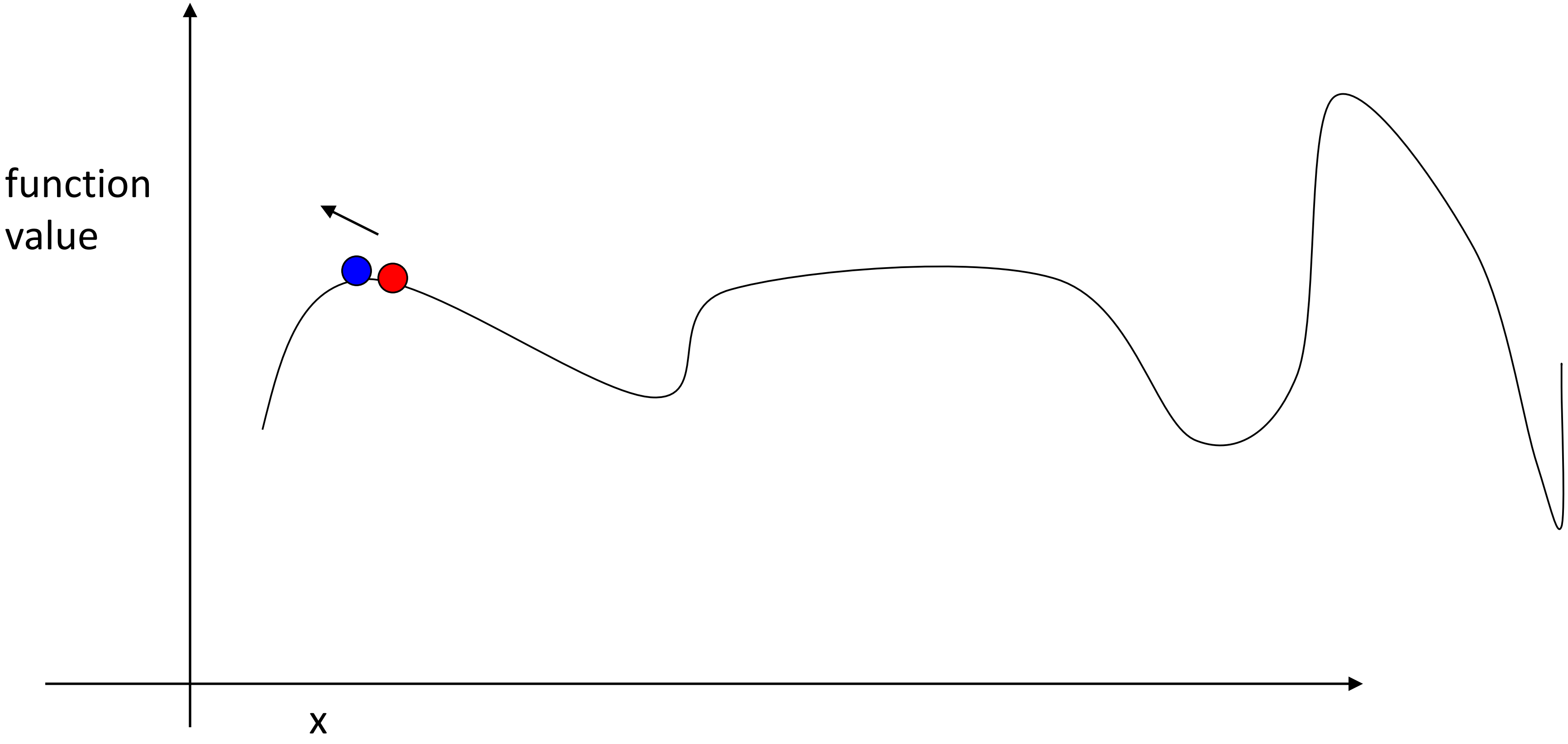
# Gradient Ascent

- Next step is actually lower, so stop



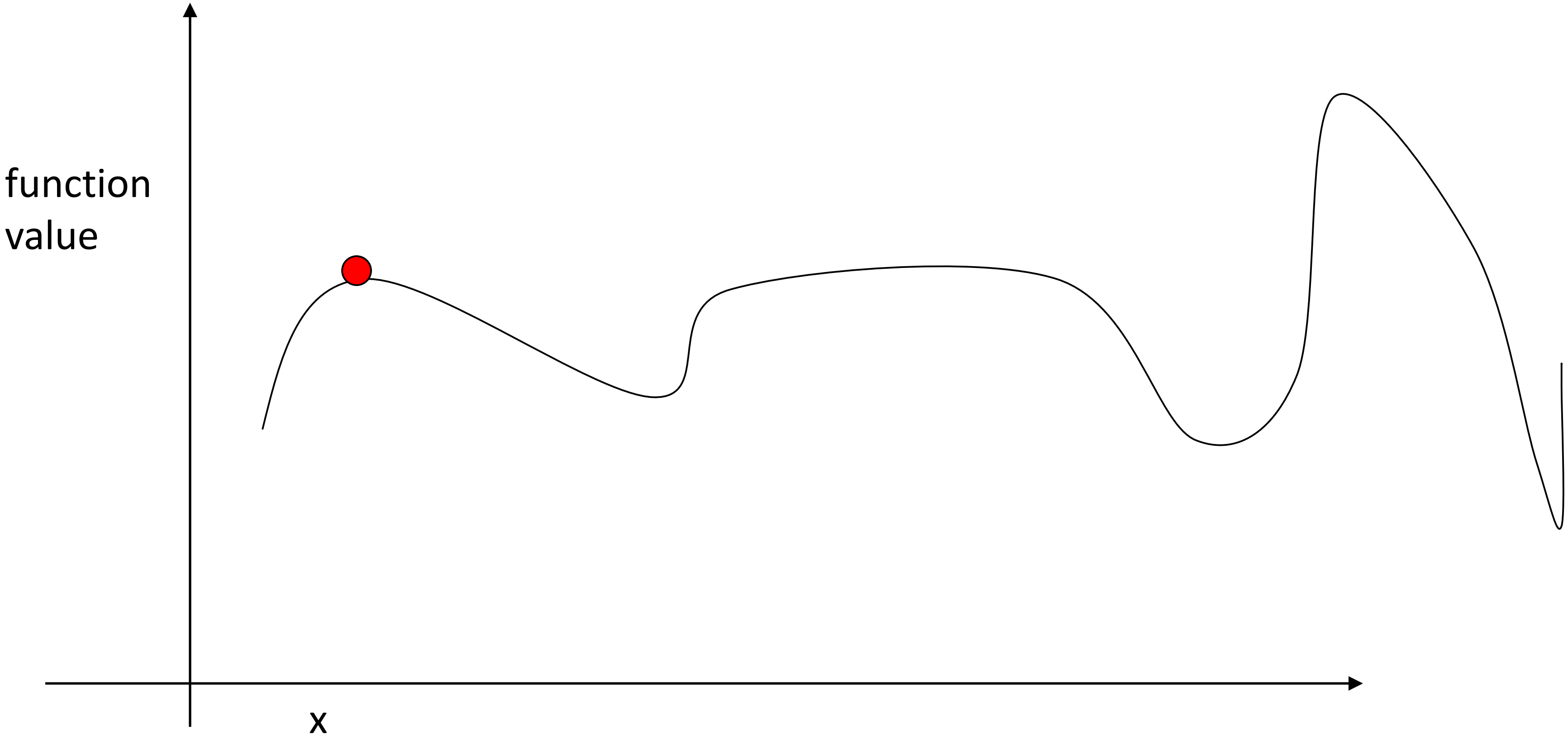
# Gradient Ascent

- Could reduce step size to “hone in”



# Gradient Ascent

- Converge to (local) maximum



# Dealing with Local Minima

- Can use various modifications of hill climbing and gradient descent
  - Random starting positions – choose one
  - Random steps when maximum reached
  - Conjugate Gradient Descent/Ascent
    - Choose gradient direction – look for max in that direction
    - Then from that point go in a different direction
- Simulated Annealing

# Nontraditional Optimization Algorithms:

They are found to be potential search and optimization algorithms for complex engineering optimization problems

I. Genetic Algorithm (GA)

II. Simulated Annealing (SA)

# Genetic Algorithm

First developed by John Holland in 1970’s

- Popularized largely by the work of David Goldberg
- Explosive growth in
  - Engineering design
  - Machine Learning/Artificial Intelligence
  - Data Mining
  - Model parameter fitting

# GA…..

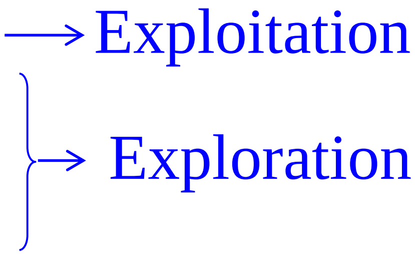
- Fall under the category of evolutionary computation
- Search and Optimization algorithms that mimic natural evolution
- Based on two fundamental principles
  - Exploration (variation)
  - Exploitation (survival of the fittest)

# GA terminology

- Chromosome: consists of a binary string used to represent designs
  - Genes: individual binary variables
  - Alleles: individual binary variable values
- Genotype: *k*ary representation of *ith* design
- Phenotype: Expressed Genotype
  - Parameter space
  - Fitness space

# GA Mechanics

- Search point (solution) represented by chromosome
- GAs work by manipulating population of chromosomes
- Three fundamental operators
  - Selection
  - Crossover
  - Mutation



Exploitation

Exploration

# GA Mechanics contd.

- Crossover and Mutation result in sampling over the solution space
- Selection needs a criterion
  - This is the objective function (fitness function in GA jargon)
    - Computer evaluates this function
    - Human makes decision (subjective function)
    - Co-evolved against predators and prey

# GA Operators

- Selection
  - Darwinian survival of the fittest.
  - Better individuals have more children (more copies in mating pool)
  - Ways to do:
    - Roulette wheel or proportionate selection
    - Tournament selection (w/w.o. replacement)
    - Truncation

# GA Operators

- Crossover
  - Combine bits and pieces of good parents
  - Speculate on, new, possibly better children
  - By itself, a random shuffle on chromosome
  - With population becomes a sampling of possible combinations (good for exploiting inherent solution patterns…)

# GA Operators

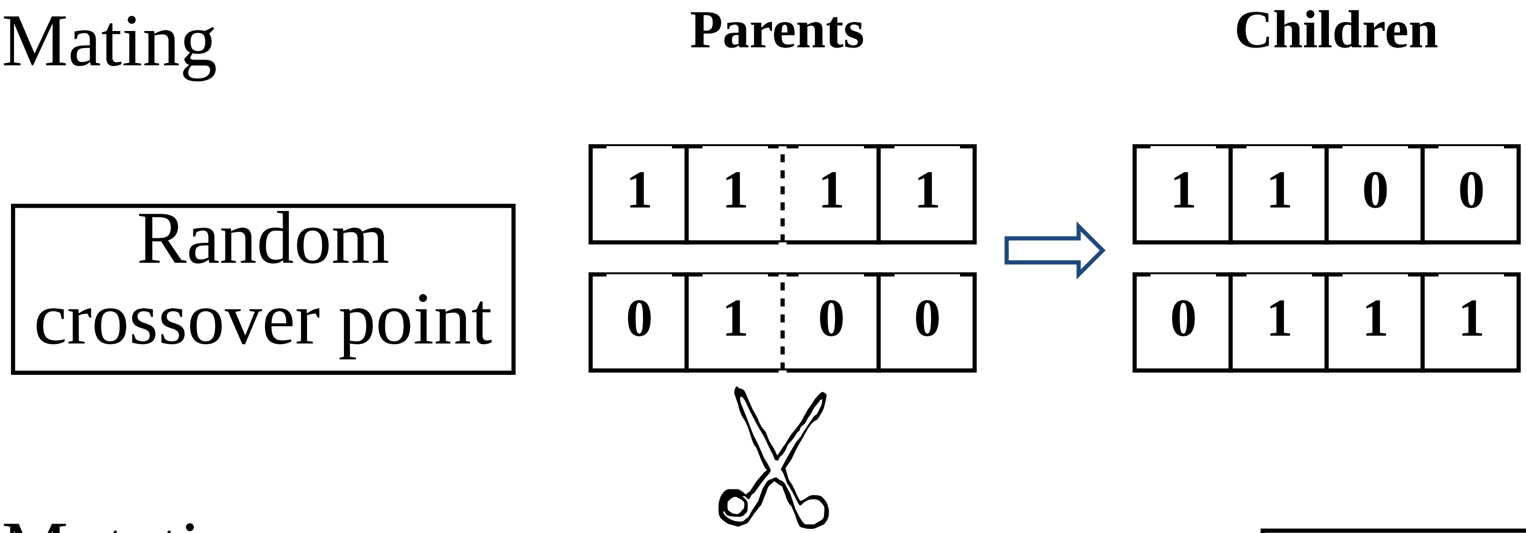
- Mutation
  - Mutation is a random alteration of a string
  - Change a gene, small movement in the neighborhood
  - By itself, a random walk
  - Helpful to prevent premature convergence
  - Normally allow at least one mutation in each population (pm= 1/n)

# GA Operators

Selection

Population members compete to mate & pass traits

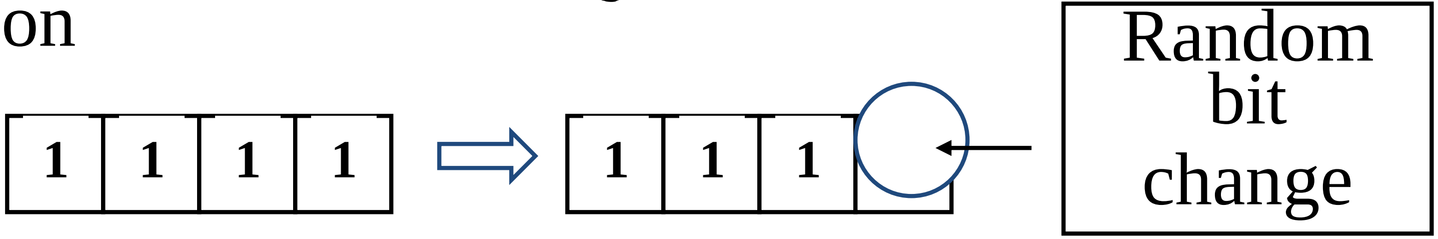
Mating



1 1 1 1 → 1 1 0 0

0 1 0 0 → 0 1 1 1

Mutation



1 1 1 1 → 1 1 1

# GA flowchart

- Start with Initial Population
- Evaluate fitness of individuals
- Select better individuals form mating pool
- Members of mating pool interact forming children
- Mutate child population
- Replace parent with child
- Repeat till ‘convergence criterion’ met

# Simulated Annealing (SA)

Simulated annealing was developed in the mid 1970's by Scott Kirkpatric.

Resembles the cooling process of molten metals through annealing

# Process of Annealing

1. Start by heating glass at high temperature, allowing molecules to move freely

2. Lower temperature of the glass slowly so that at each temperature the atoms can move just enough to begin adopting the most stable orientation.

3. Do this until temperature no longer alters glass.

# Main Ideas of Annealing

1. The temperature determines how much mobility the atoms have.

2. How slowly you cool glass is critical.
   This rate is called the *cooling schedule (aka Annealing Schedule)*.

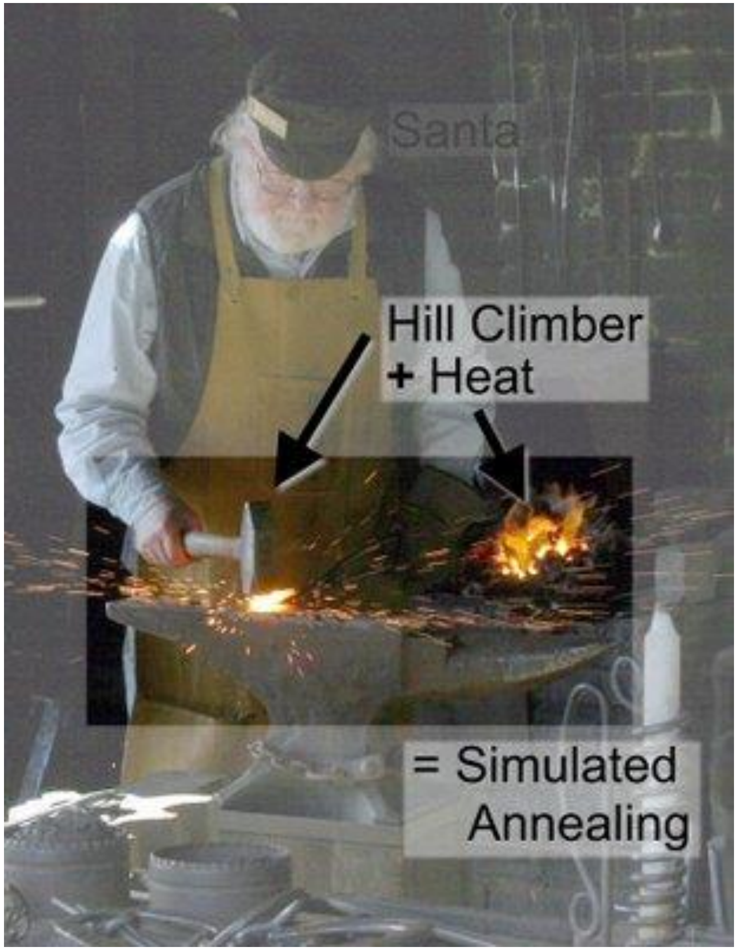
3. If the glass is cooled slowly enough,
   the atoms are able to "relax" into the most stable orientation.

# What is *Simulated* Annealing?

- Simulated Annealing (SA) uses the same ideas as regular annealing

- It is a *Metropolis Monte Carlo (MMC) Optimization Method*

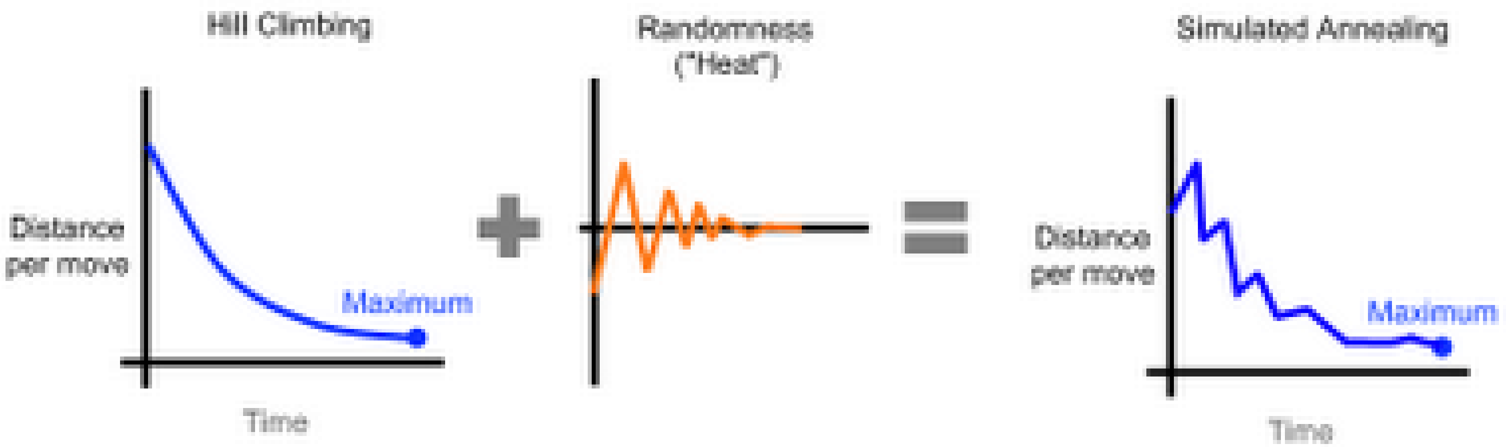
  (used especially when the global extrema are hidden amongst many local extrema).



The blacksmith's hammer guides the iron into the desired shape and density while the heat made the iron more malleable and responsive to the hammer. Essentially hill climbing is equivalent to a blacksmith hammering without heat.

Source:www.maxdama.com/2008/07/trading-optimization

# Sim. annealing is hill climbing plus the addition of random jumps.



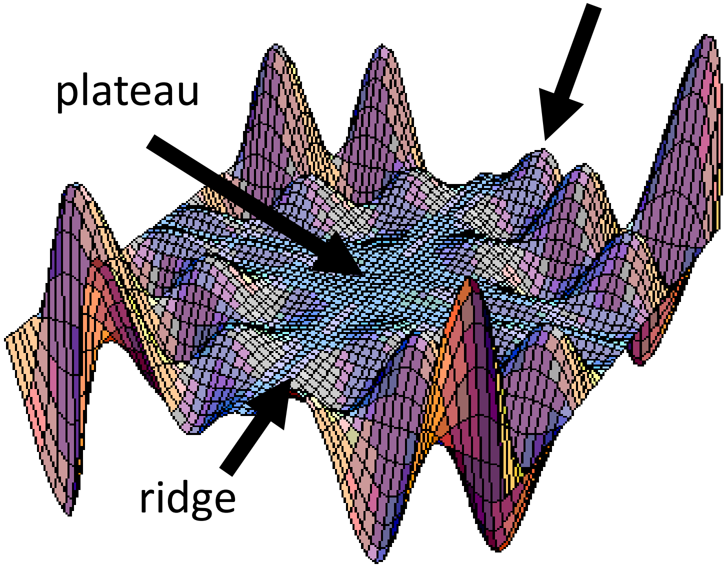
Source:www.maxdama.com/2008/07/trading-optimization

# Exploring the Landscape

- **Local Maxima**: peaks that aren't the highest point in the space

- **Plateaus:** the space has a broad flat region that gives the search algorithm no direction (random walk)

- **Ridges:** flat like a plateau, but with drop-offs to the sides; steps to the North, East, South and West may go down, but a step to the NW may go up.



local maximum

plateau

ridge

Image from: http://classes.yale.edu/fractals/CA/GA/Fitness/Fitness.html

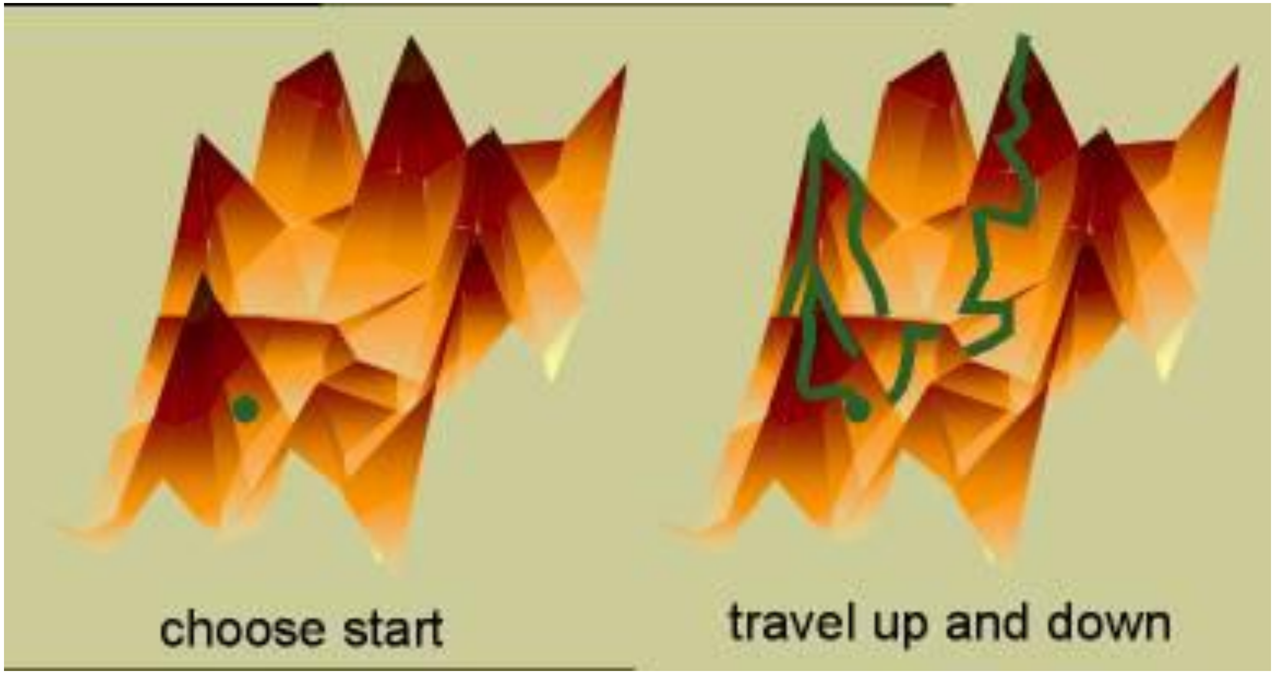
# Drawbacks of hill climbing

- Problems: local maxima, plateaus, ridges
- Remedies:
  - **Random restart:** keep restarting the search from random locations until a goal is found.
  - **Problem reformulation:** reformulate the search space to eliminate these problematic features
- Some problem spaces are great for hill climbing and others are terrible.

# SA intuitions

- combines hill climbing (efficiency) with random walk (completeness)
- Analogy: getting a ping-pong ball into the deepest depression in a bumpy surface
  - shake the surface to get the ball out of the local minima
  - not too hard to dislodge it from the global minimum
- Simulated annealing:
  - start by shaking hard (high temperature) and gradually reduce shaking intensity (lower the temperature)
  - escape the local minima by allowing some “bad” moves
  - but gradually reduce their size and frequency

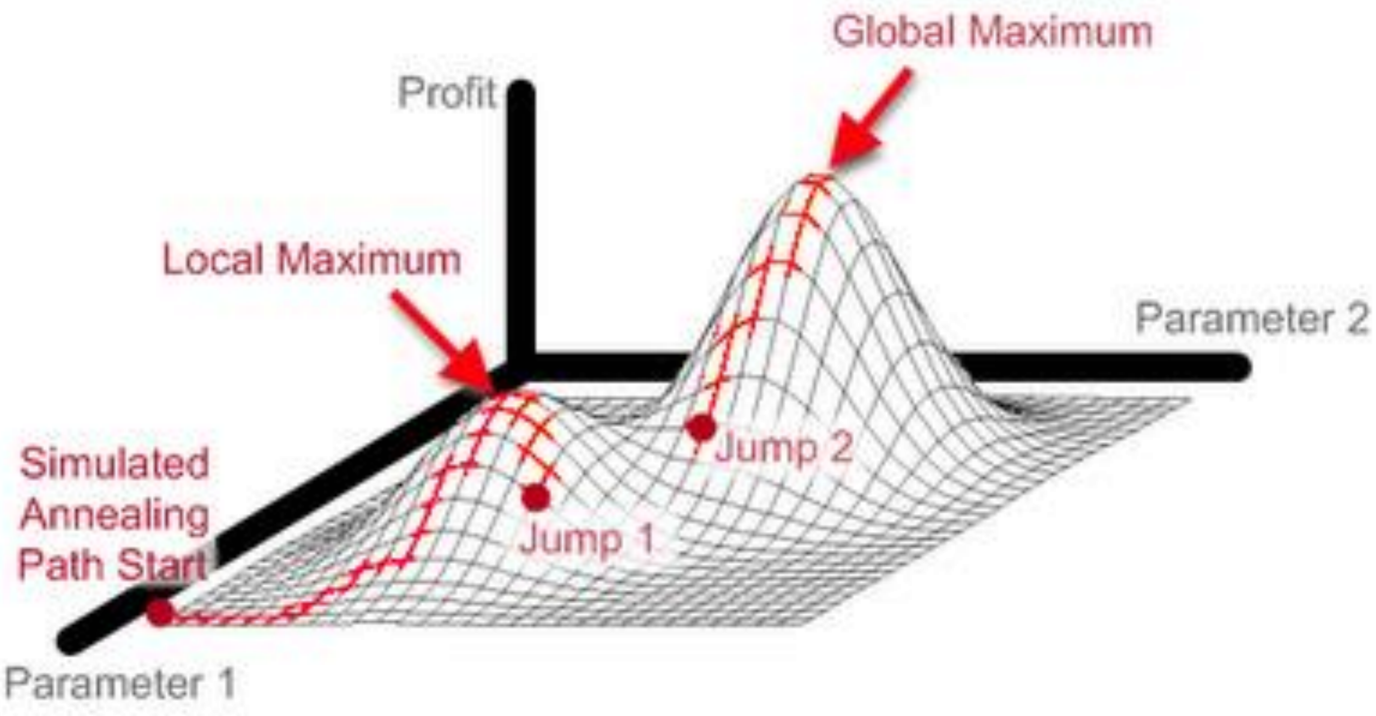
# Search Process



Source:biology.st-andrews.ac.uk/.../simanneal.png

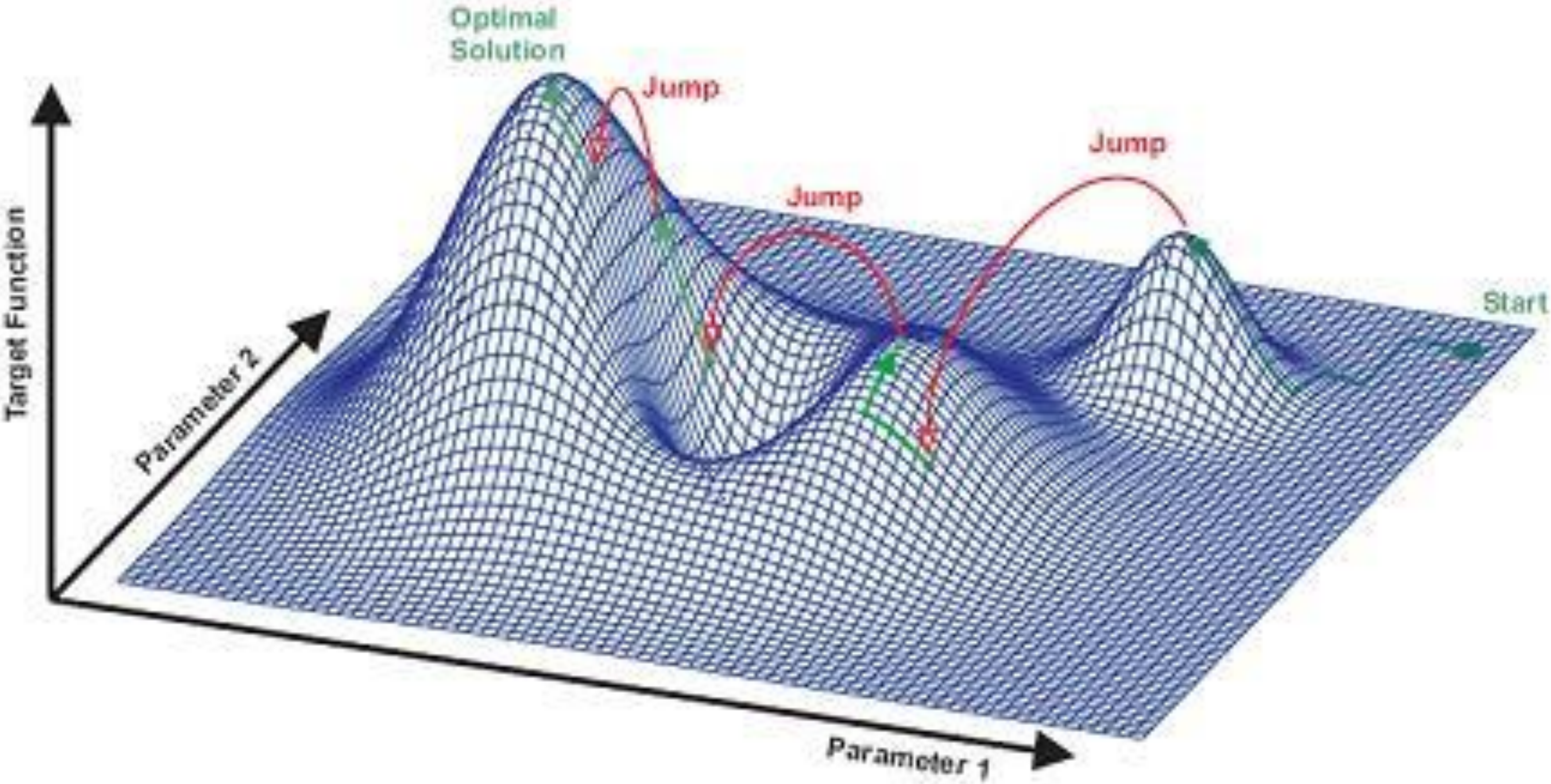
Simulated annealing will outperform hill climbing when the local maximum is near the global maximum. In this case one of the jumps may get far enough from the local max to reach the ascending slope of the global max:

Simulated Annealing can escape local minima with chaotic jumps



Source:www.maxdama.com/2008/07/trading-optimization

# Simulated Annealing



# Linchpin of SA

- During an MMC iteration, if at current guess for minima X
- Function value goes down, ACCEPT X.
- Function value goes up, ACCEPT X with probability

$$P(E(t+1)) = \min [1, \exp (-\Delta E/kT)]$$

- where:
  T is the temperature of the system
  k is the Boltzmann Constant (Metropolis)
  $\Delta E$ is the difference in function value from the previous iteration

# SA algo..

***Step 1***

*Choose an initial point* $x^0$

*Termination criterion* $\varepsilon$

*Set Temperature T high and*
*Set n, and t = 0*

***Step 2***

*Calculated randomly created neighborhood point*

$x^{t+1} = N(x^t)$

***Step 3***

*If* $\Delta E = E(x^{t+1}) - E(x^t) < 0$

*Else create a random number (r) in the range (0,1).*

*If* $r \le \exp(\Delta E / T)$ *Set t=t+1;*

*Else go to Step 2*

***Step 4***

*If* $\left| x^{t+1} - x^t \right| < \varepsilon$ *and T is small,*
***Terminate;***

*Else if (t mod n)=0 then T= RT.T*

*Go to Step 2*

*Else go to Step 2*

# AN EXAMPLE
# on
# GROUNDWATER MONITORING

# Groundwater Monitoring

- Selection of sampling points and
- Temporal sampling frequency

To determine:

physical, chemical, and
biological characteristics of groundwater.

# Motivation

- Designing effective and efficient LTM plans can be difficult, especially at sites with any of the following:
  - Many wells
  - Many possible constituents that could be sampled
  - Different types of samples with different costs and effectiveness
    - Traditional well-based samples
    - Indicator samples
    - Sensor data
- Mathematical optimization can help in identifying effective plans

# Why Optimize?

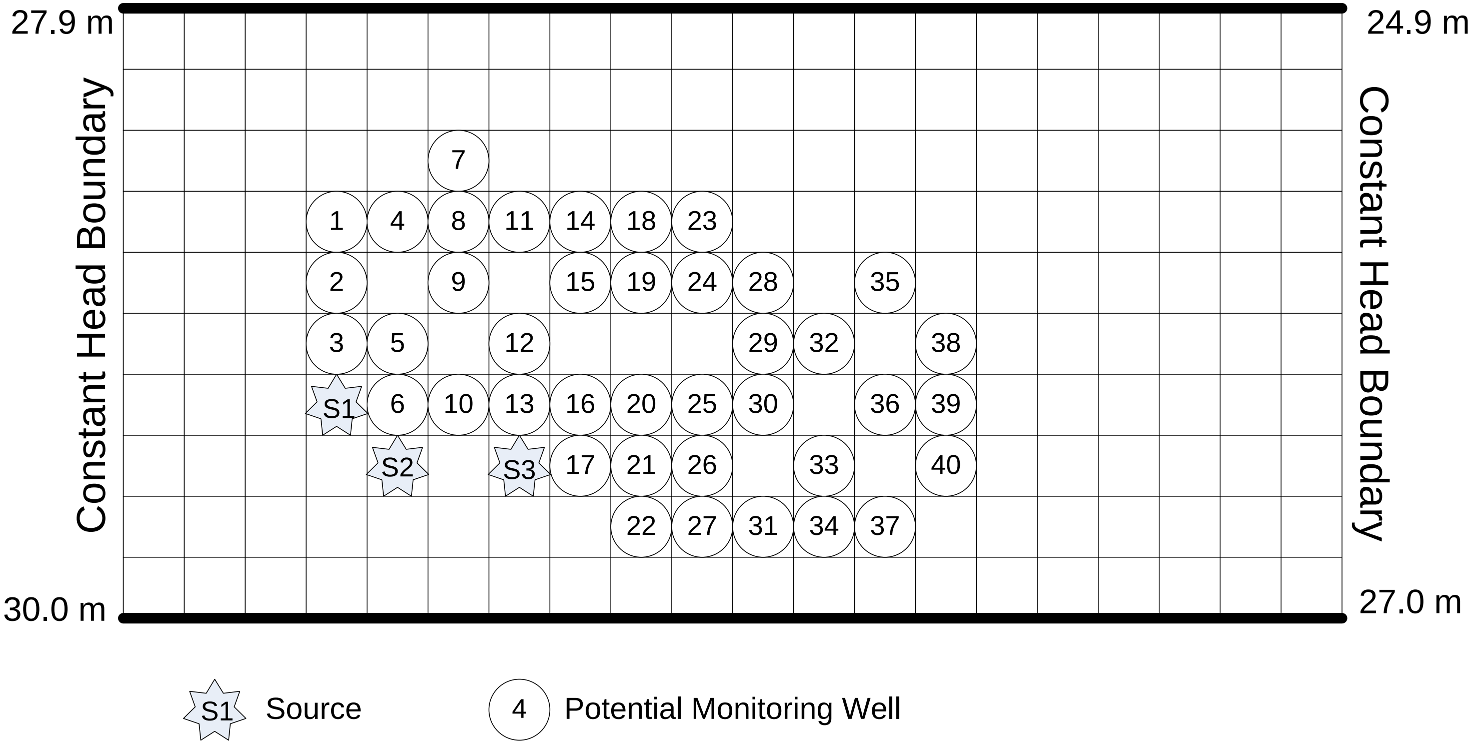
- The number of possible sampling plans in one monitoring period is

$$2^m$$

where m is the product of

  - Number of possible sampling locations
  - Number of constituents at each well
  - Number of types of samples

# Study Area for Analysis



# Why Optimize (Contd.)?

- For example, a site with 10 wells and 3 possible constituents to measure at each well would have

possible sampling plans $2^{30}$ or 1 billion!

- Any trial-and-error method is unlikely to identify the most effective sampling plans
- Mathematical optimization can efficiently identify the most effective sampling plans to satisfy any monitoring objective that can be quantified

# Components of an optimization formulation

- **Decision Variables:**

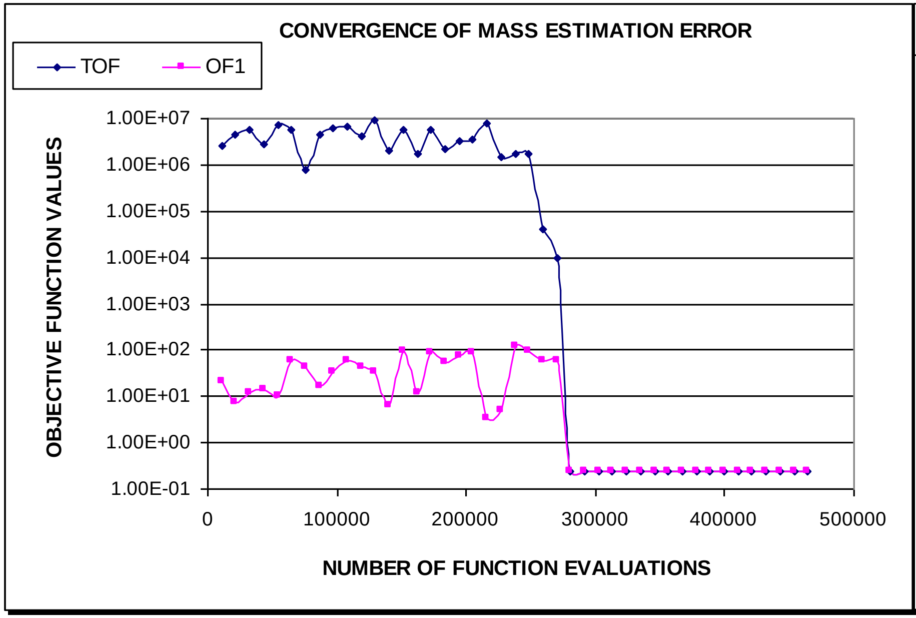
  What we are determining optimal values for?

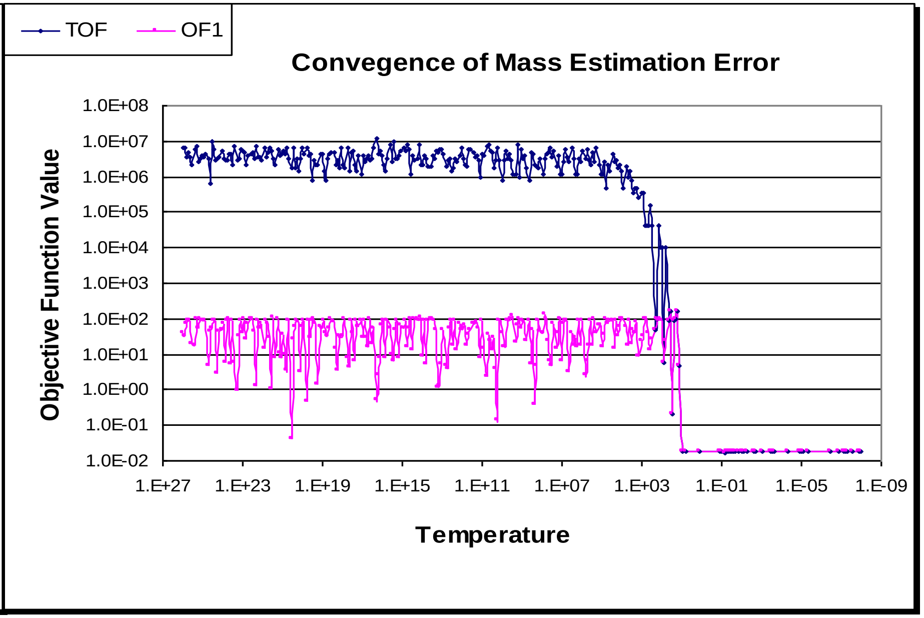
- **Objective function:**

  1. Values can be computed once the value of each decision variable is specified.
  2. The mathematical equation being minimized or maximized.
  3. Serves as the basis for comparing one solution to another.

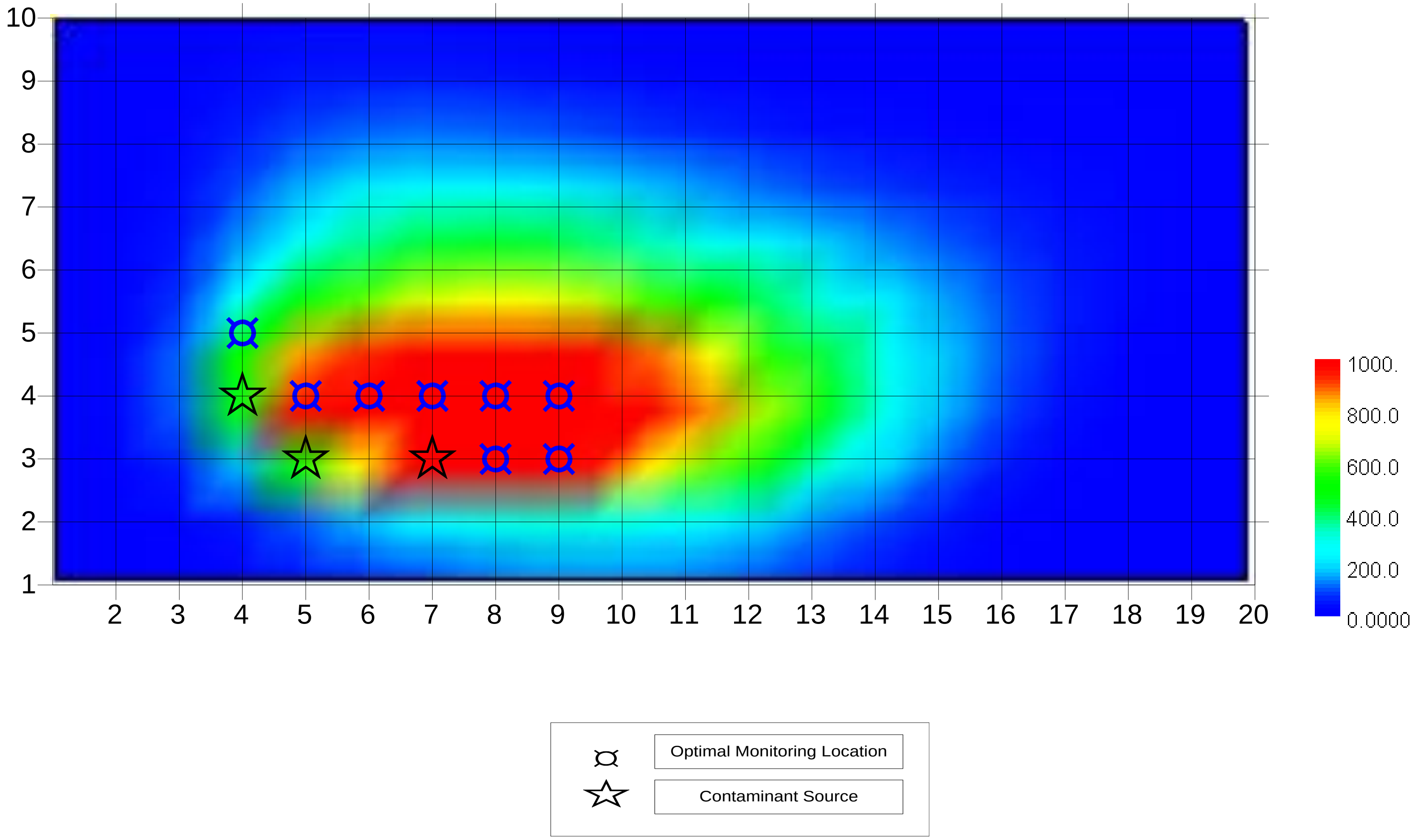
- **Constraints:** Limits on values of the decision variables, or limits on other values that can be calculated once the value of each decision variable is specified.

# Sensitivity Analysis



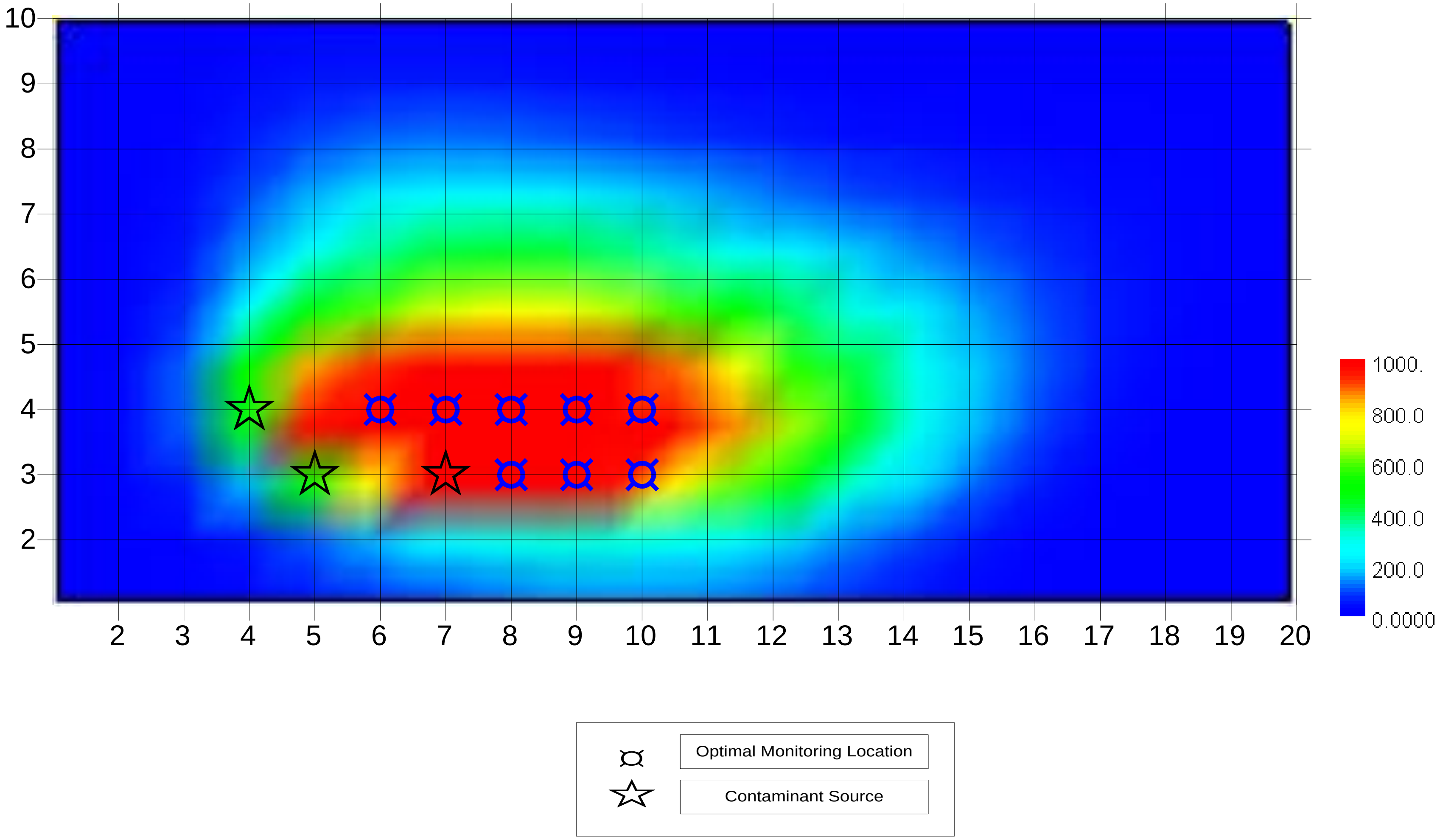


# management period I (GA)



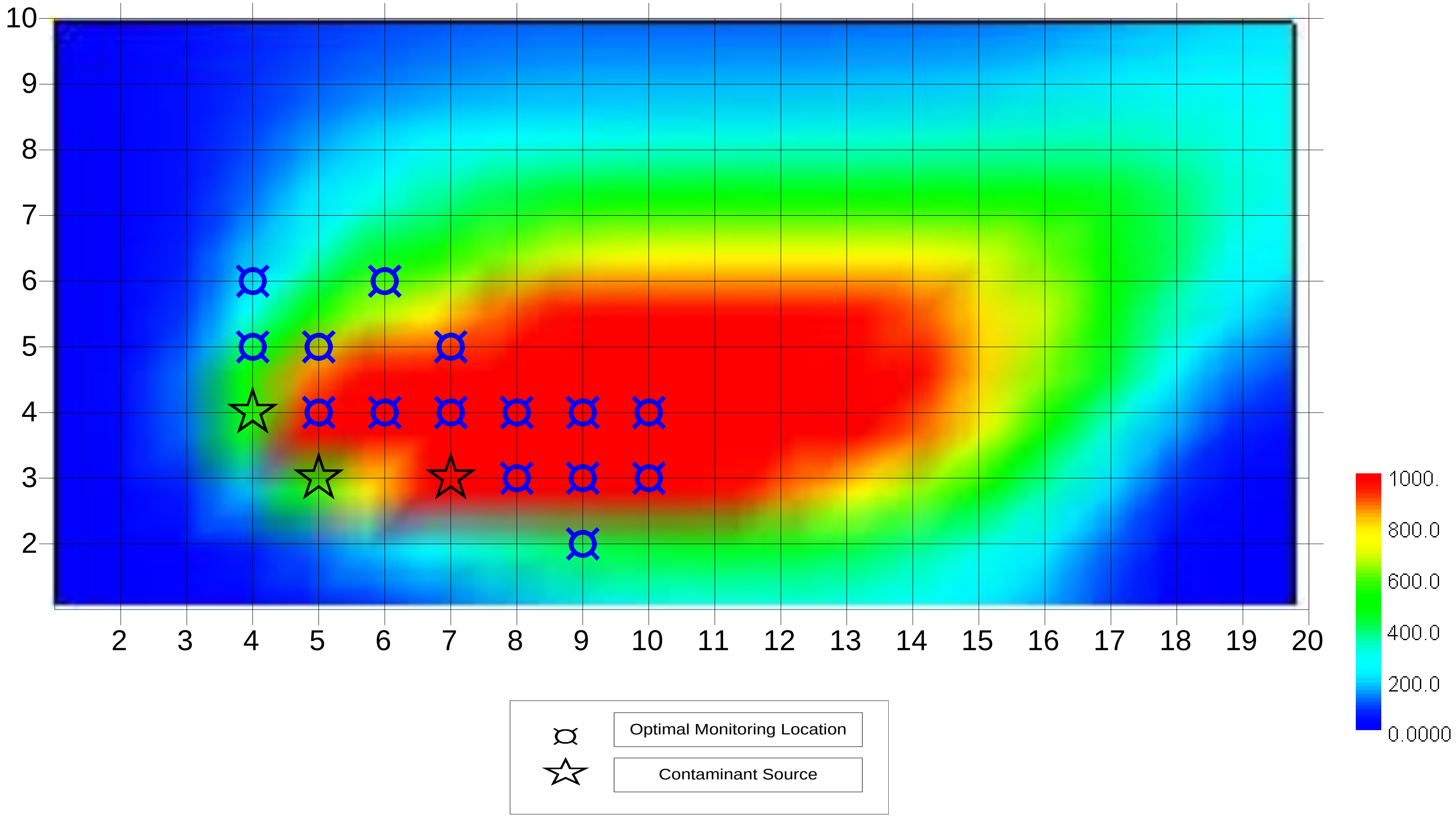
Optimal location of 8 wells

# management period I (SA)



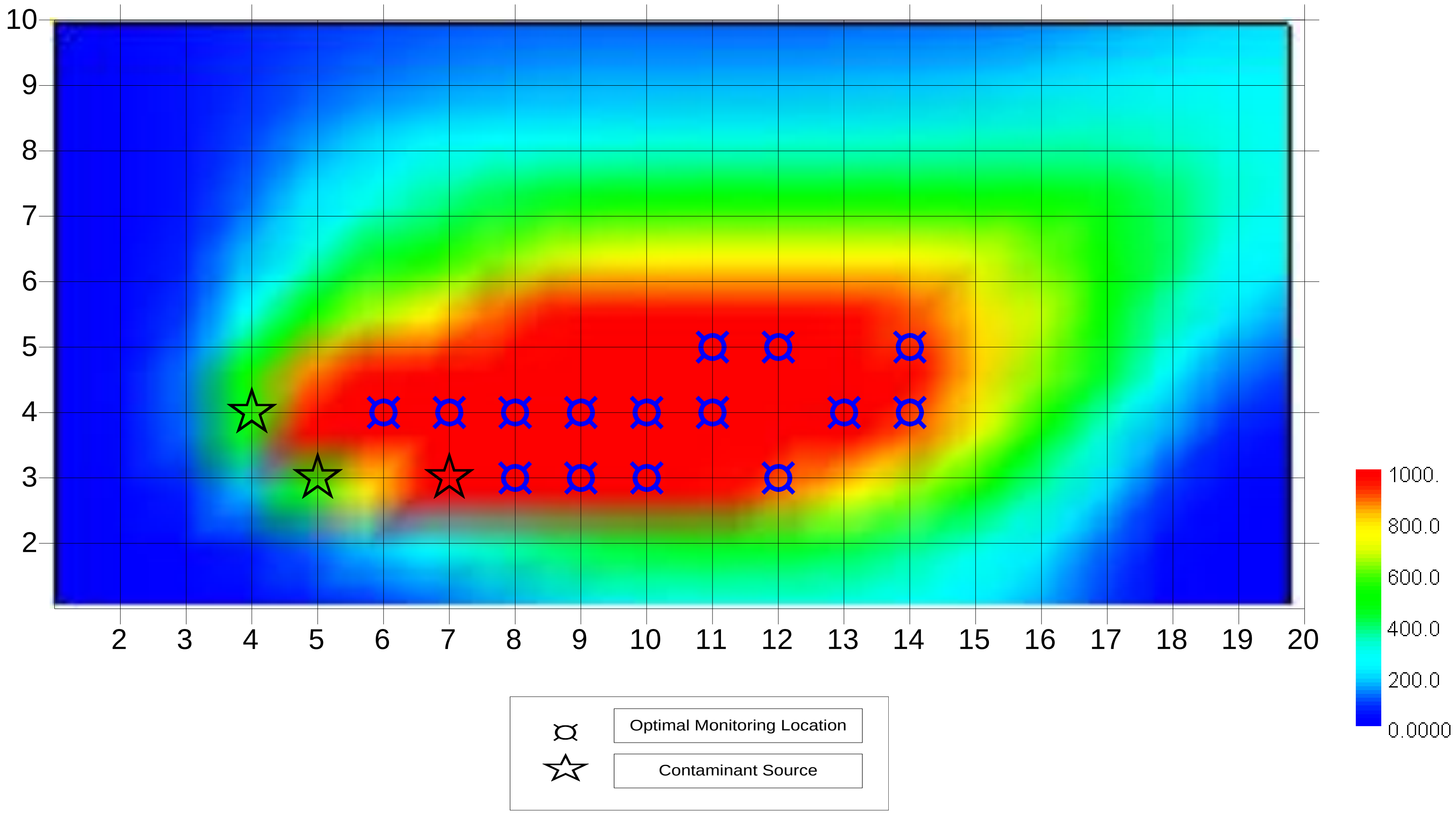
Optimal location of 8 wells

# management period II(GA)



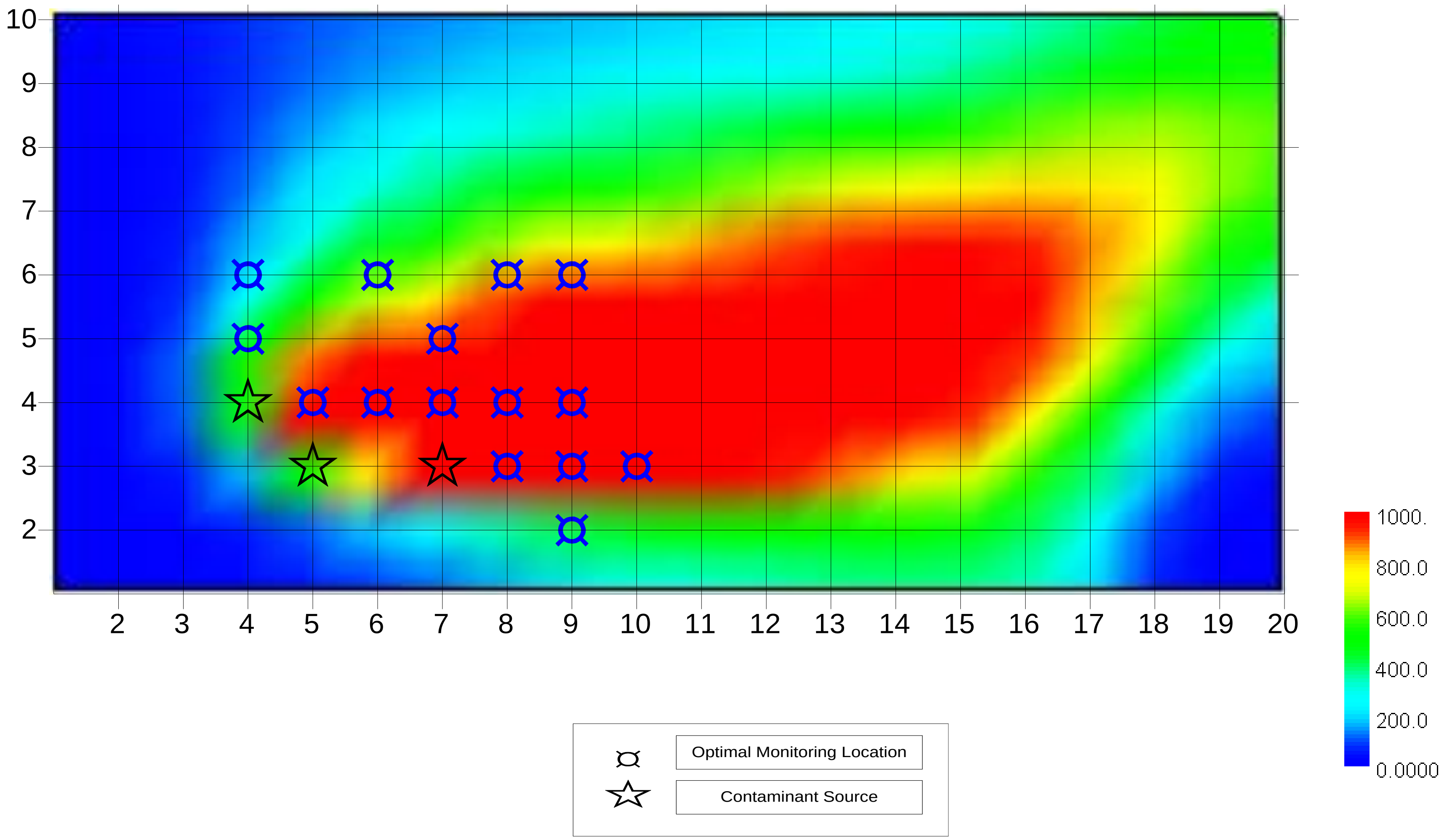
Optimal location of 15 wells

# management period II (SA)



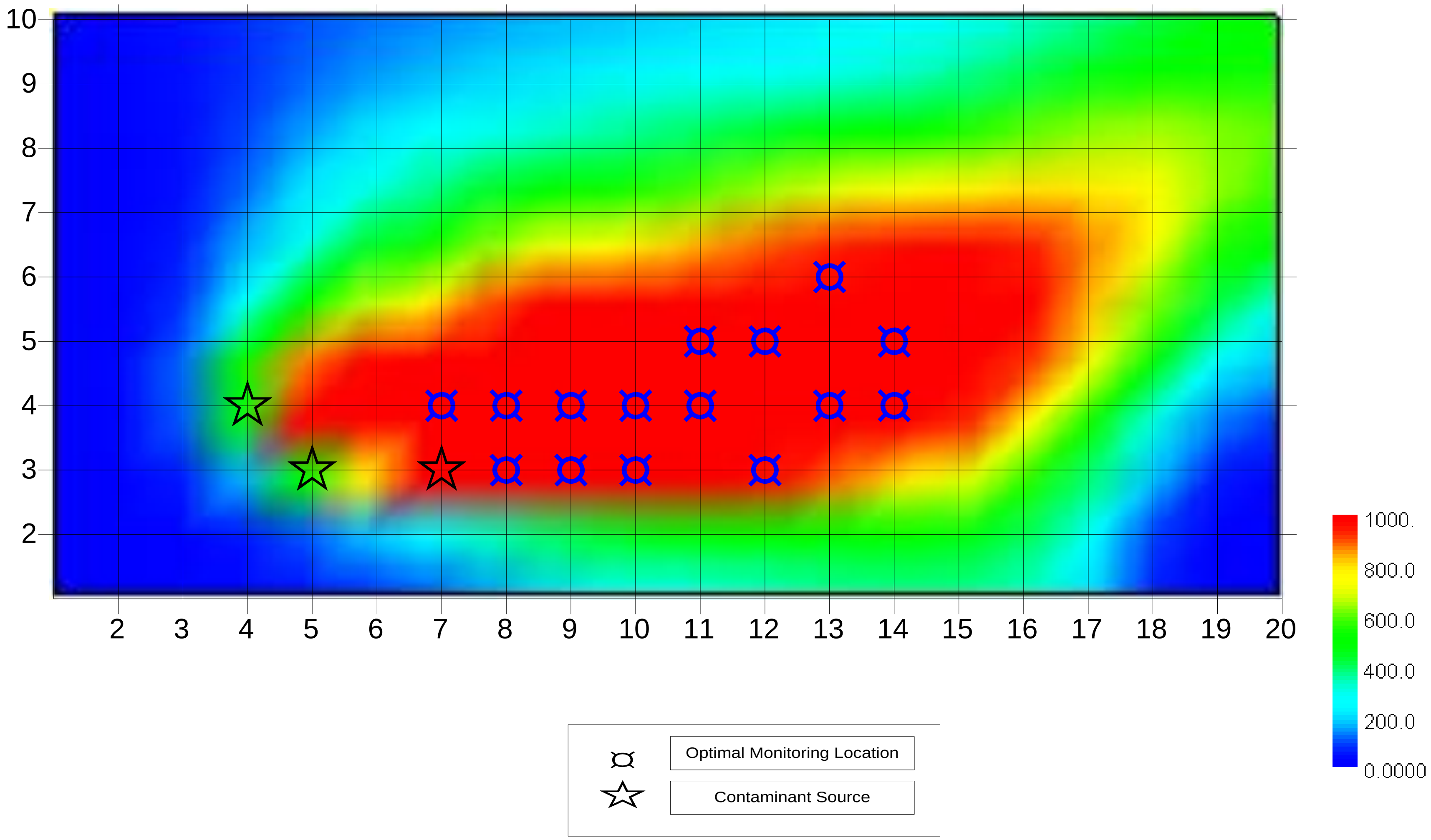
Optimal location of 15 wells

# management period III (GA)



Optimal location of 15 wells

# management period III (SA)



Optimal location of 15 wells

# GA & SA

- The genetic process of the standard algorithm uses a "crossover" as do chromosomes in living organisms.
- For two parameters that are close to each other in the coding, a child is likely to get both from one of its parents. While for parameters that are far apart, the child is likely to get one from each parent.
- If the parameters have been transformed to binary, then some pairs of bits should usually inherit from the same parent -- but the algorithm will not know which ones those are.
- If no transformation has been made, then each location should inherit independently of all others.

# GA & SA

- In SA algorithm one solution is held at a time, and random steps away from this solution are taken.
- If the random step results in a better solution, then that becomes the new solution about which random steps are taken.
- As the optimization proceeds, the average size of the steps gradually decreases.
- The simulated annealing algorithm accepts all new points that lower the objective, but also, with a certain probability, points that raise the objective.
- By accepting points that raise the objective, the algorithm avoids being trapped in local minima, and is able to explore globally for better solutions

Source: (http://www.burns-stat.com).

# GA & SA

- With SA, one usually talks about *solutions*, their *costs*, and *neighbors* and *moves*;
- While with GA, one talks about *individuals* (or *chromosomes*), their *fitness*, and *selection*, *crossover* and *mutation*.
- SA can be thought as GA where the population size is only one. The current solution is the only individual in the population. Since there is only one individual, there is no crossover, but only mutation.

# Conclusions

- SA and GA are quite close relatives, and much of their difference is superficial.
- The problem of groundwater monitoring is framed as Integer programming and SA outperforms GA.
- The scope of SA is still to be explored

  for various other engineering application.

Thank You