OPTIMIZATION PROBLEMS AND ALGORITHMS

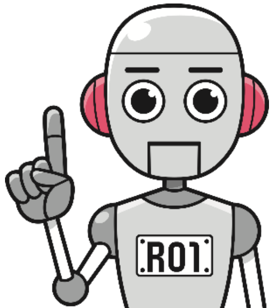
Assoc. Prof. **Seyedali Mirjalili**

Director of the Centre for Artificial Intelligence Research and Optimisation

Torrens University Australia

ali.mirjalili@laureate.edu.au, https://seyedalimirjalili.com/

AIRO: Centre for Artificial Intelligence Research and Optimisation



Outlines

- Optimization problems
  - Components
  - Inputs
  - Constraints
  - Objectives

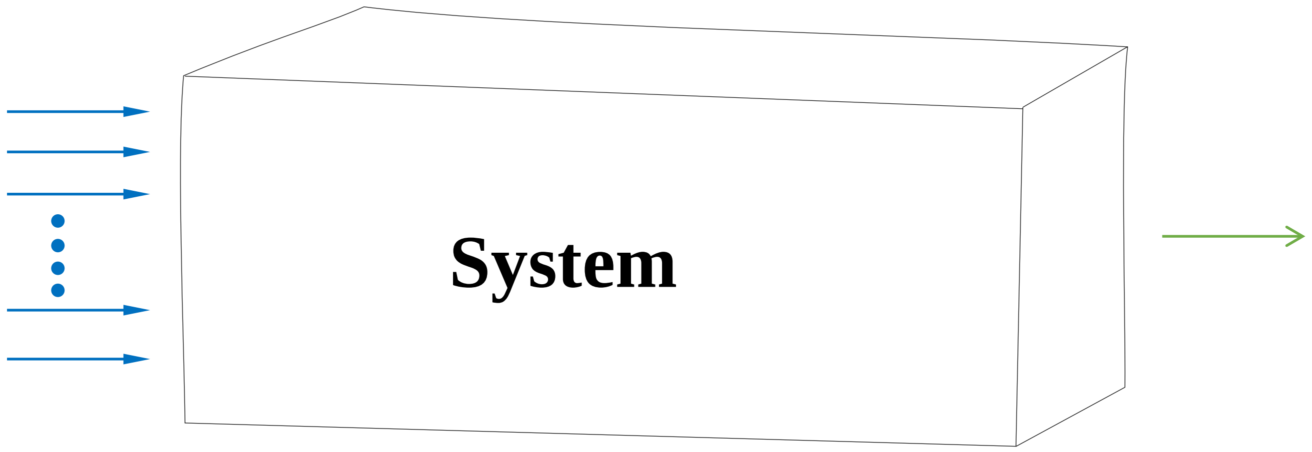
- Optimization algorithms
  - Conventional
  - Modern
  - NFL theorem

OPTIMIZATION PROBLEMS

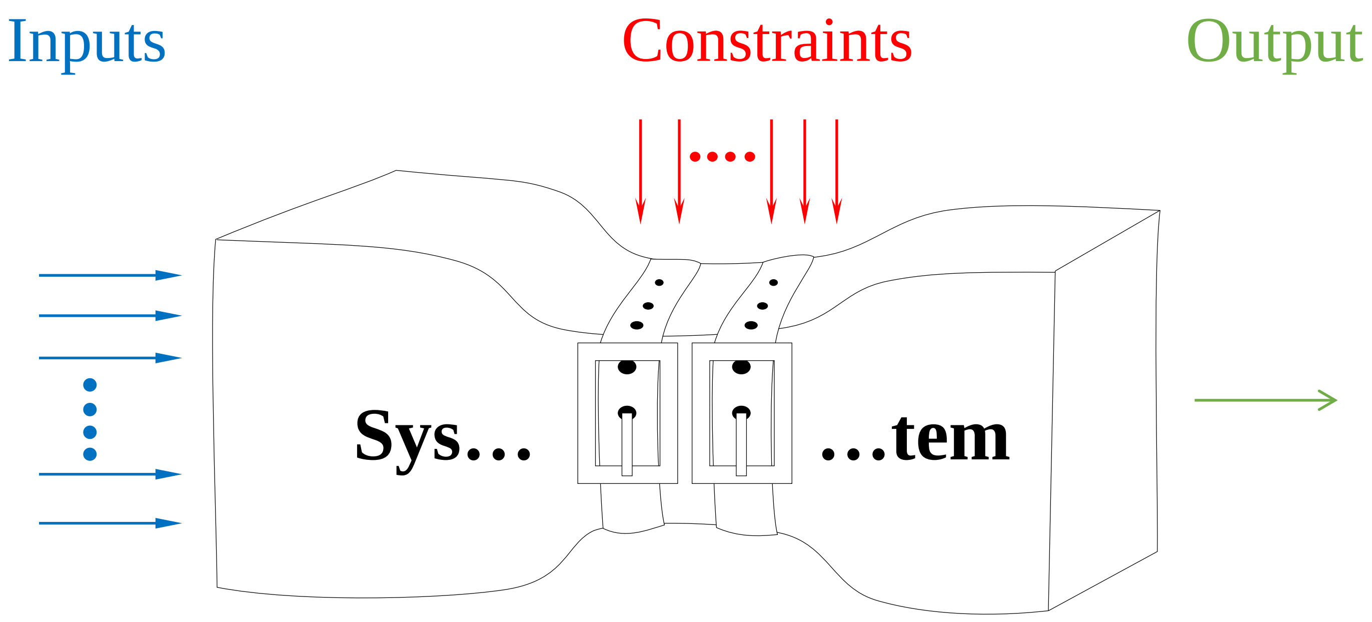
Main components of an optimization problem

Inputs (variables)

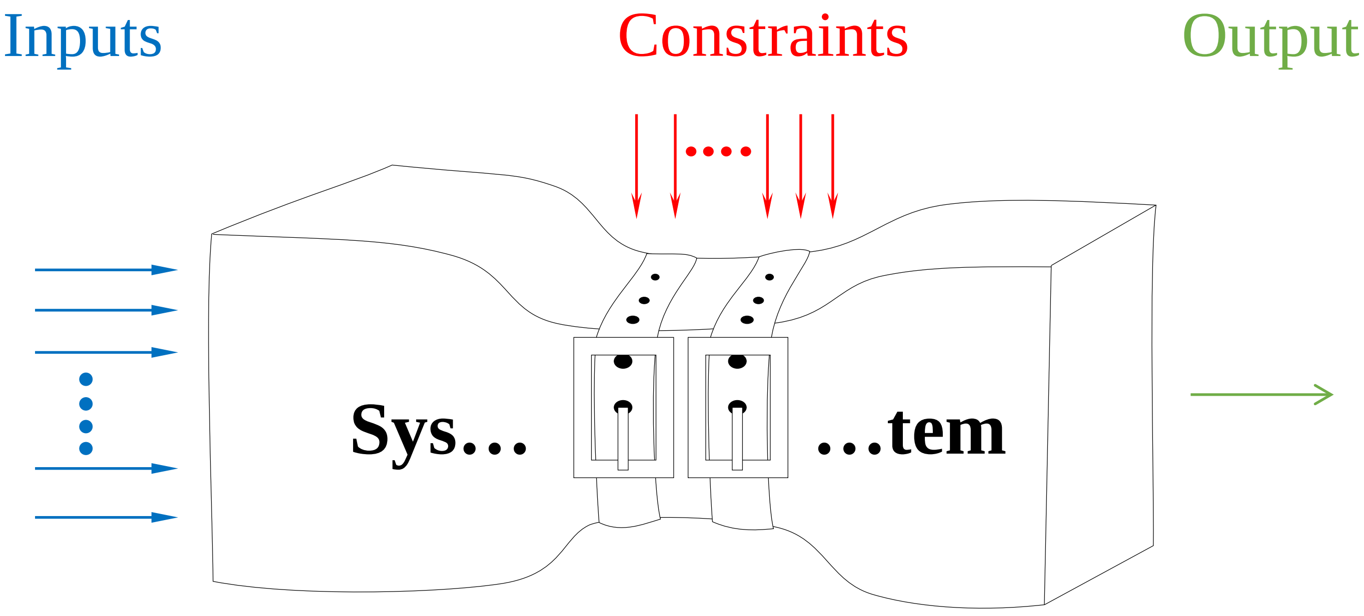
Output (objective)



Main components of an optimization problem



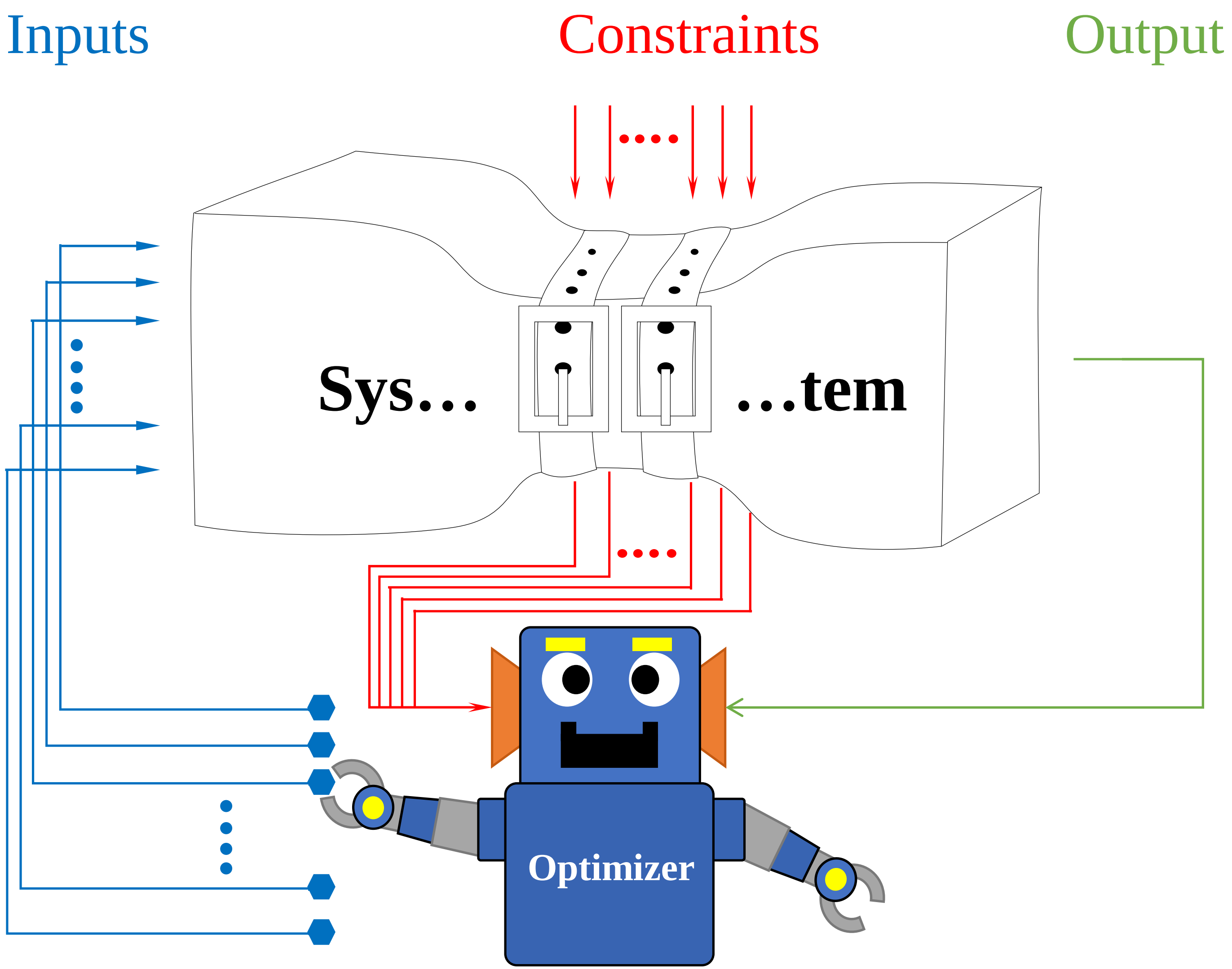
Formulating an optimization problem



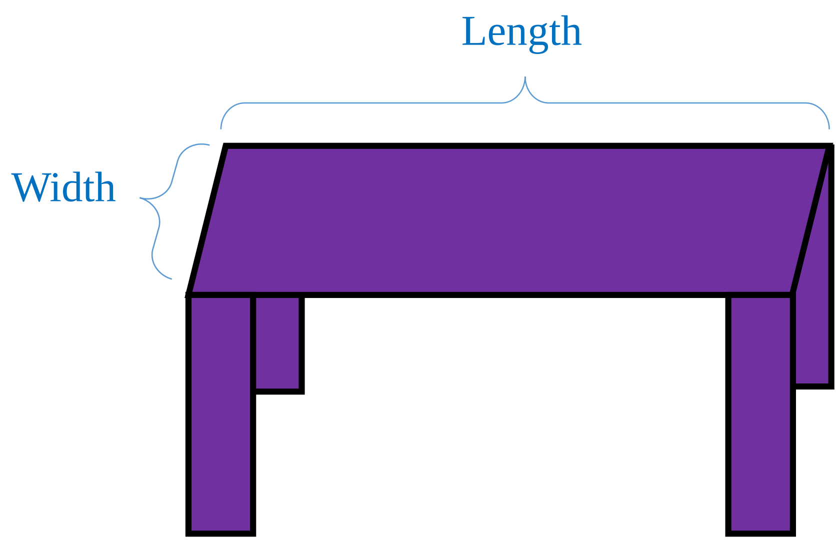
$$Minimise: \quad f(x_1, x_2, \ldots, x_n)$$

$$Suject\ to: \quad Constraints$$

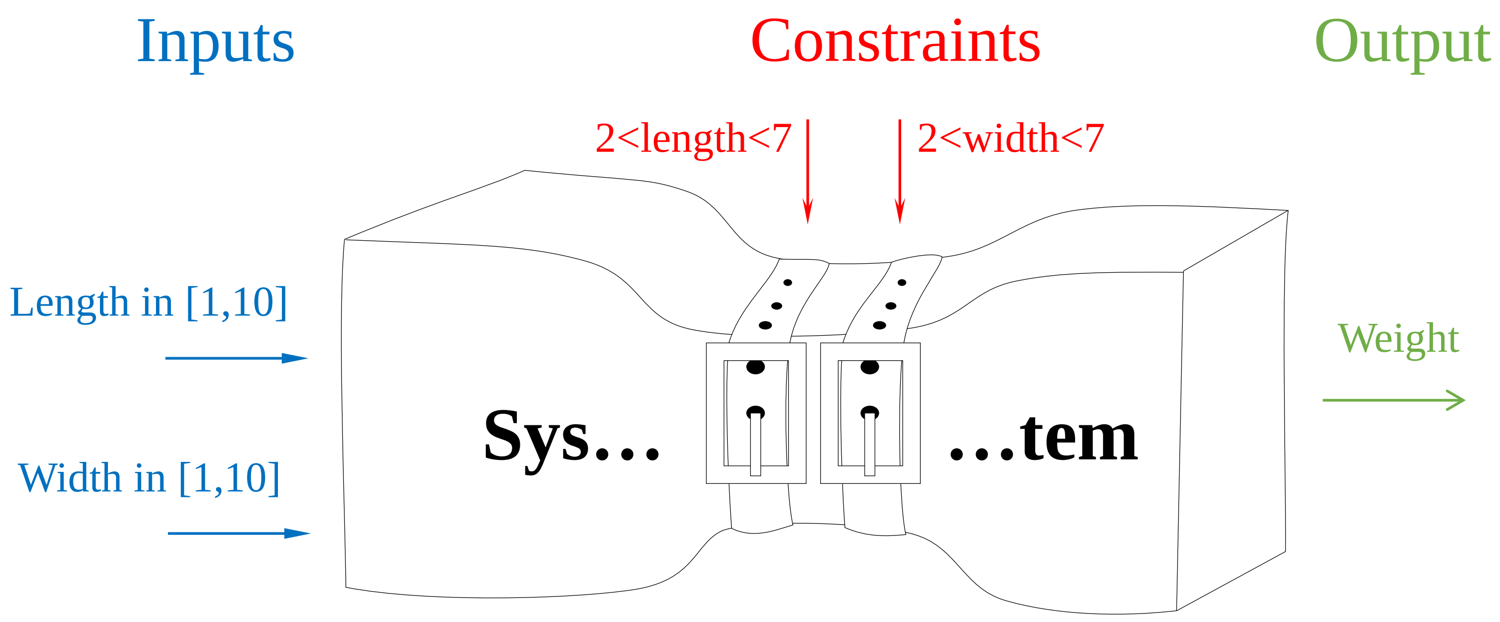
Optimization algorithm

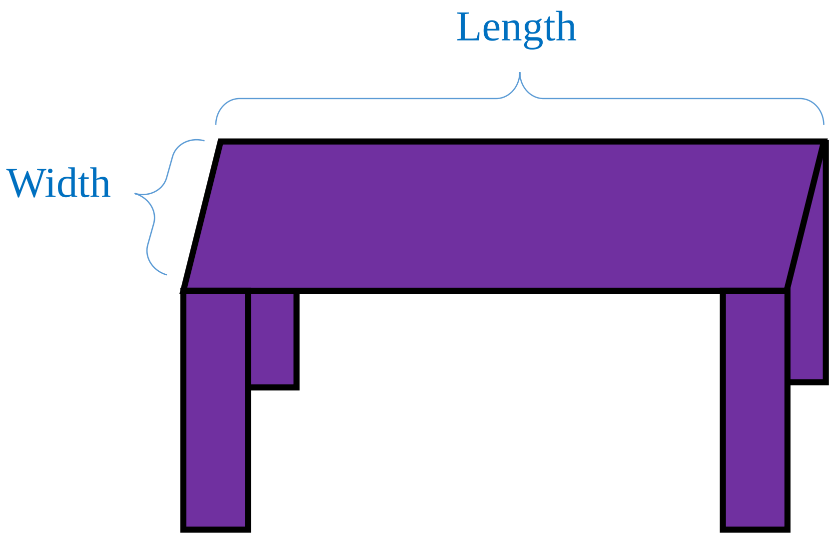


Example: designing a table



Example: designing a table

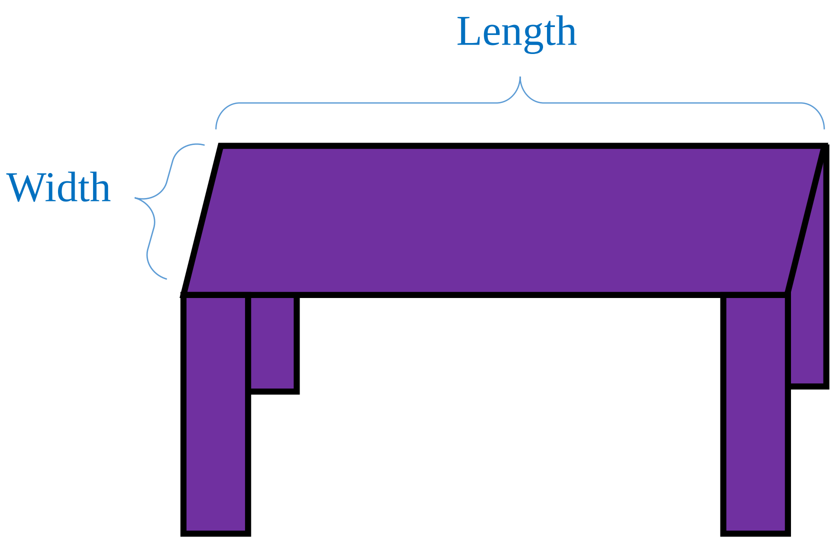




Weight = 1*(Length + Width)

The objective is to minimize the weight

Example: designing a table



$$Minimise: \quad f(length, width) = 1*(Length + Width)$$

$$Suject\ to: \quad 2 < length < 7$$
$$2 < width < 7$$

Search landscape of the table problem

Inputs:

width , length

Output:

weight

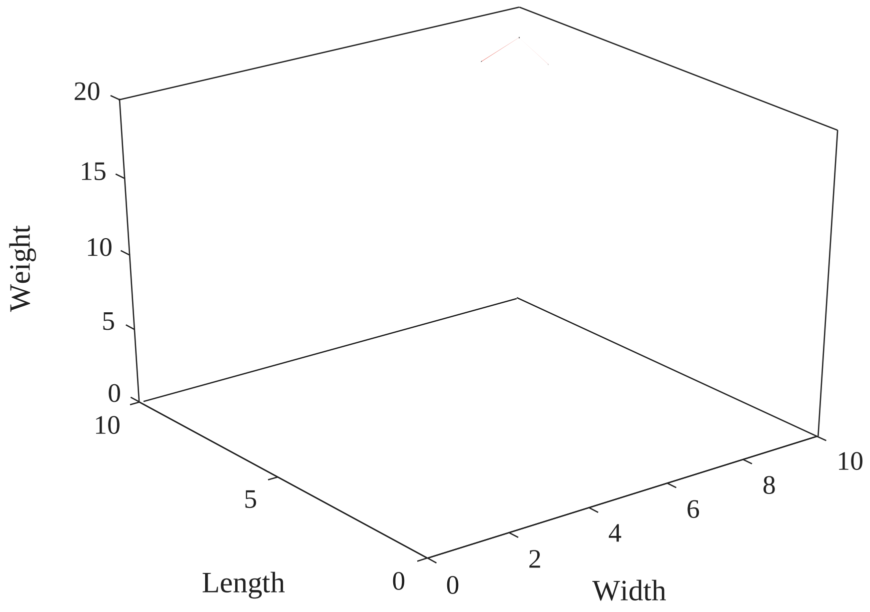


Table #1: W=10, L=10

Table #2: W=9 , L=10

Table #3: W=10, L=9

Table #4: W=9 , L=9

Search landscape of the table problem

Inputs:

width , length

Output:

weight

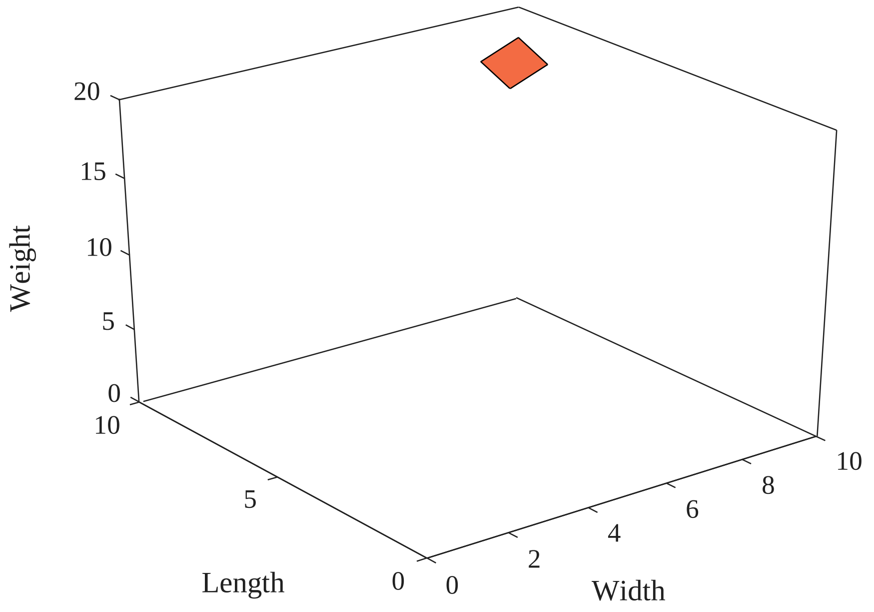


Table #1: W=10, L=10

Table #2: W=9 , L=10

Table #3: W=10, L=9

Table #4: W=9 , L=9

Example

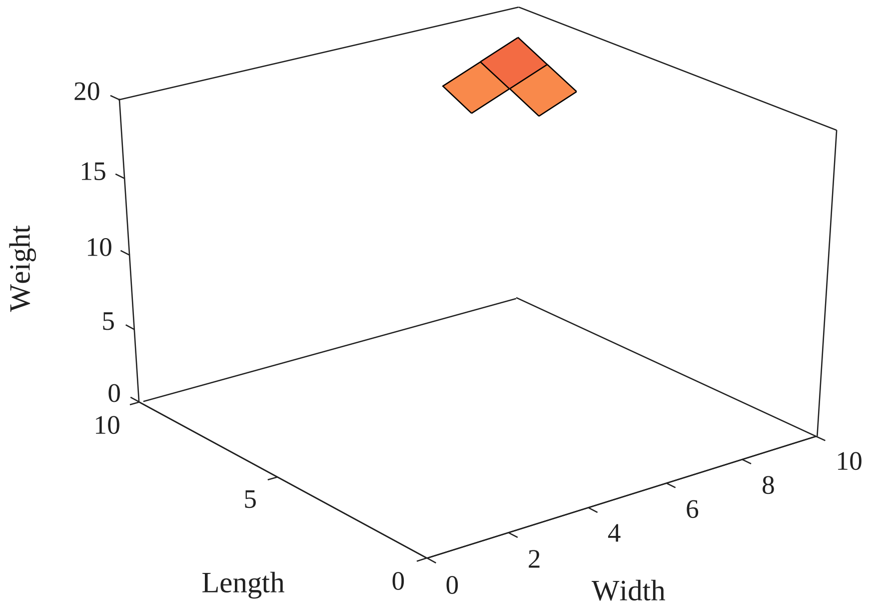
8 tables

Inputs:

width , length

Output:

weight



Example

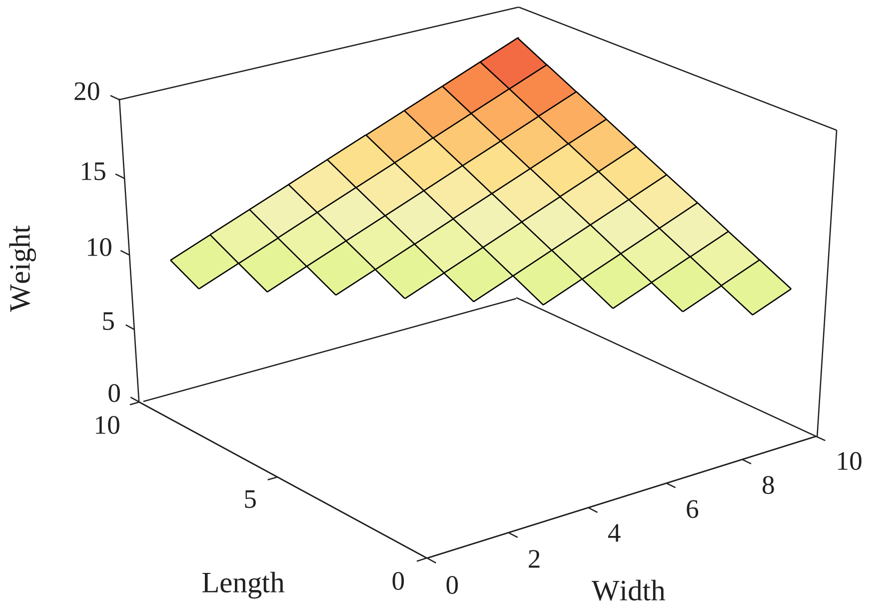
50 tables

Inputs:

width , length

Output:

weight



Example

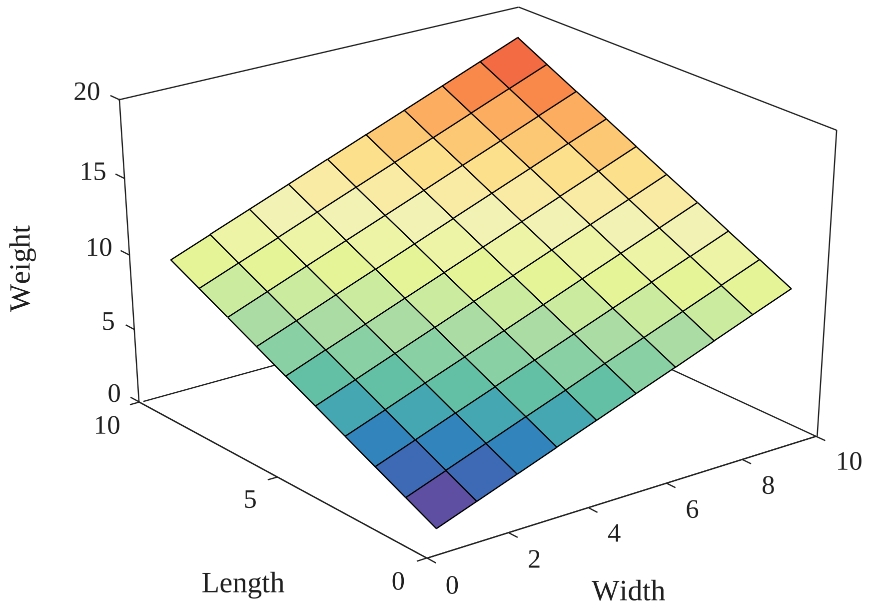
100 tables

Inputs:

width , length

Output:

weight



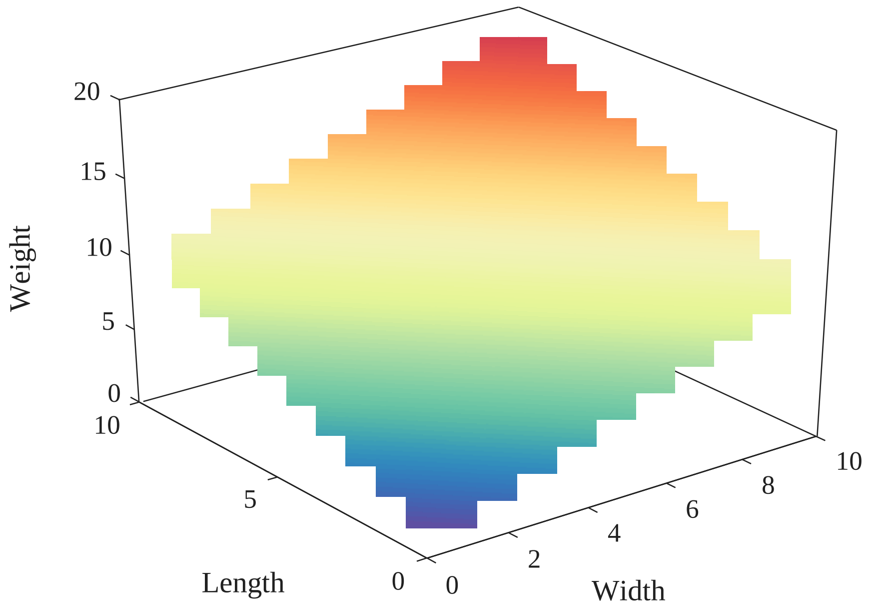
Example

Inputs:

width , length

Output:

weight



Example

Inputs:

width , length

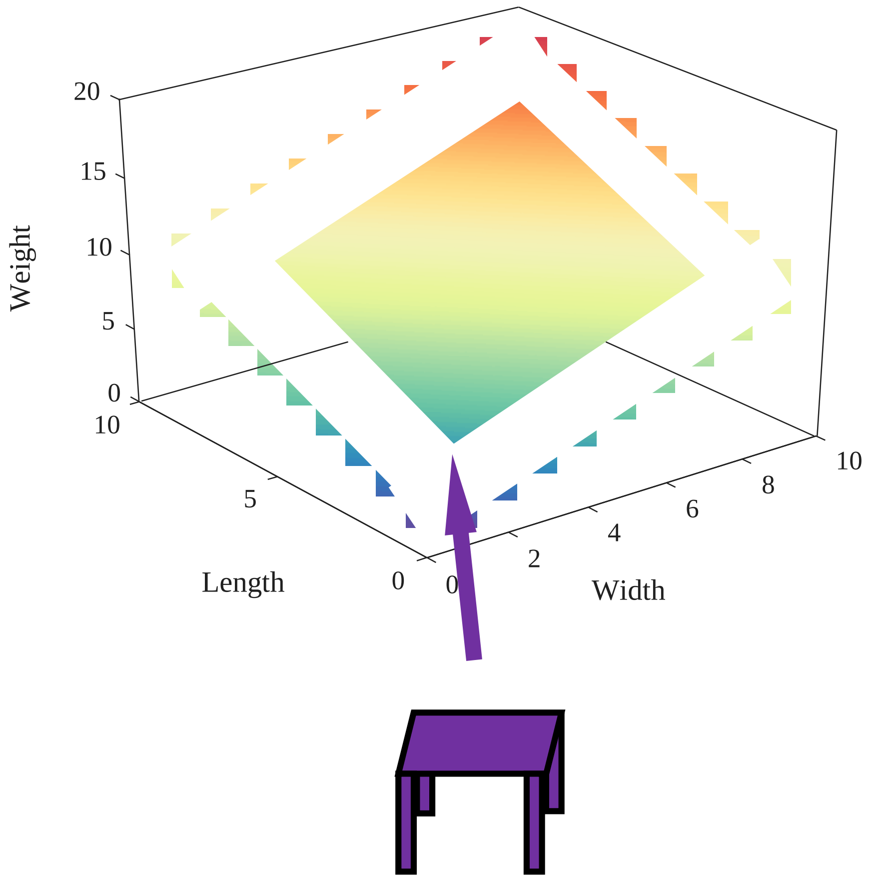
Output:

weight

Constraints:

2< width <7

2< length <7



Search landscape

Inputs:

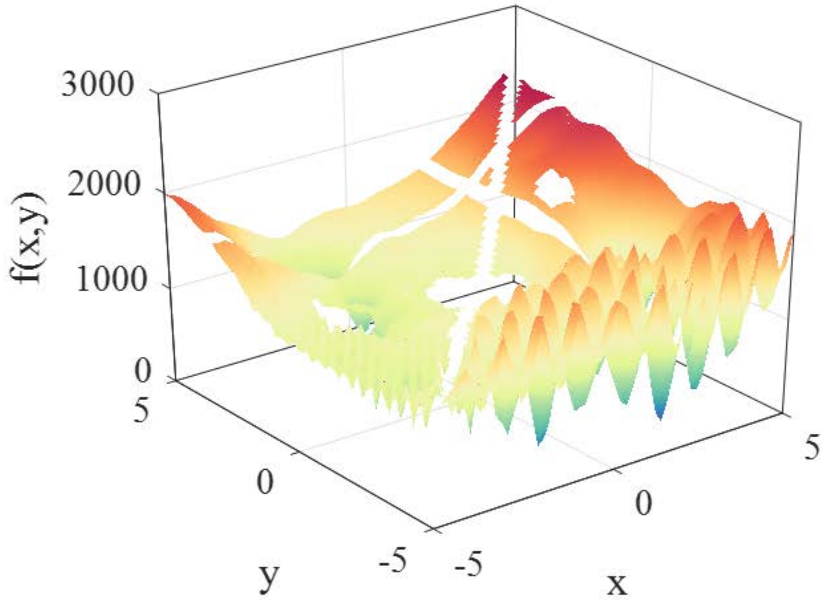
x , y

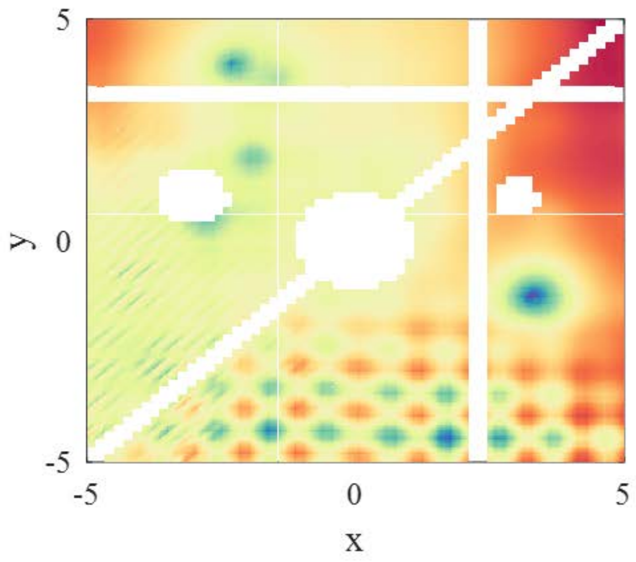
Output:

f(x,y)

Constraints:

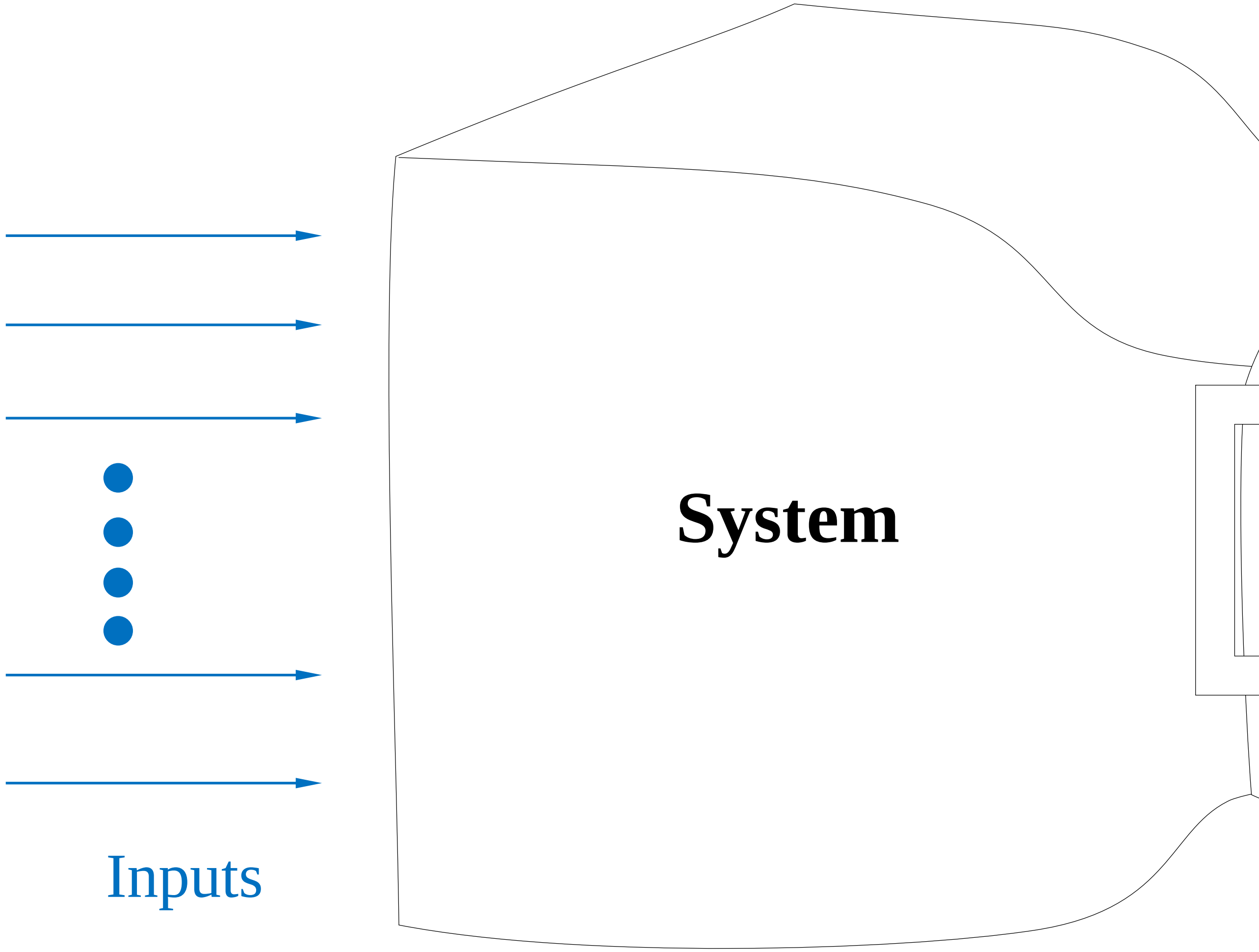
$$(y \leq 3.2) \vee (y \geq 3.4)$$
$$(x \leq 2.2) \vee (x \geq 2.3)$$
$$(x-3)^2 + (y-1)^2 \geq 0.1$$
$$(x+3)^2 + (y-1)^2 \geq 0.3$$
$$x^2 + y^2 \geq 1$$
$$x \neq y$$





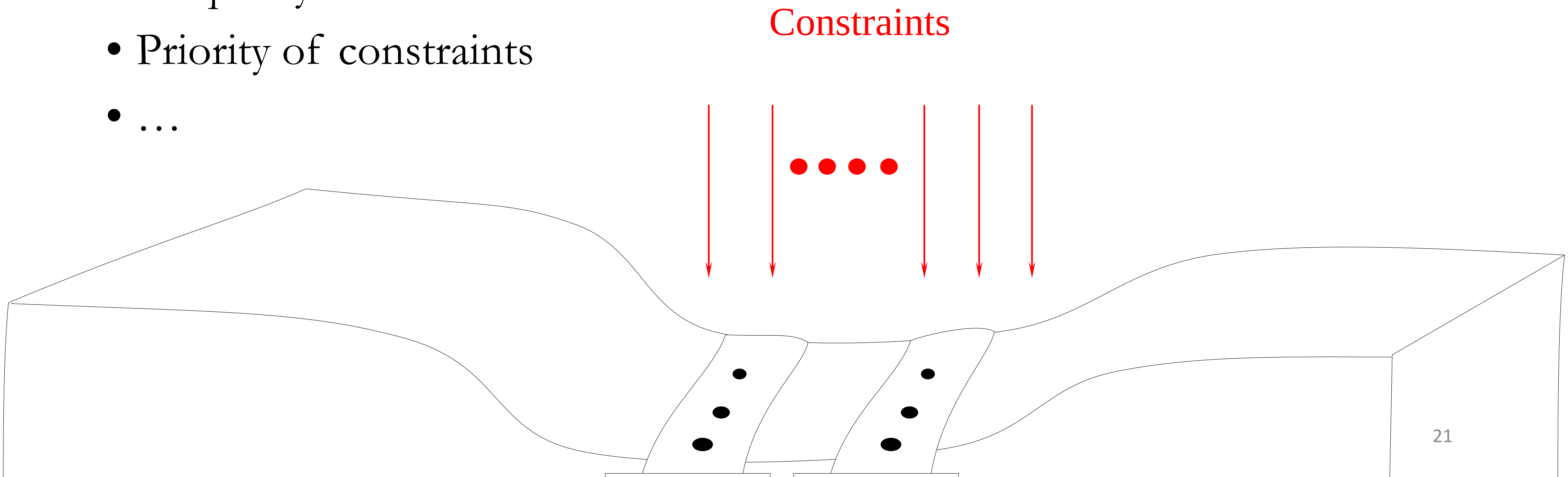
Inputs (decision variables)

- Large number of inputs
  - Large scale optimization
  - World record: 1 billion variables
- Variables with different ranges
- Dependency between the inputs
- Discrete variables
- Mix variables
- Noisy inputs
  (manufacturing errors)
- ….



Constraints

- Highly constrained landscape
- Equality constraint
- Inequality constraint
- Priority of constraints
- …



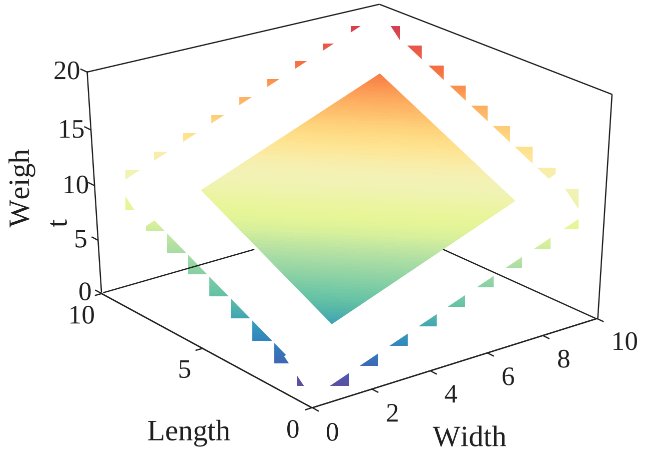
Constraints in designing a table

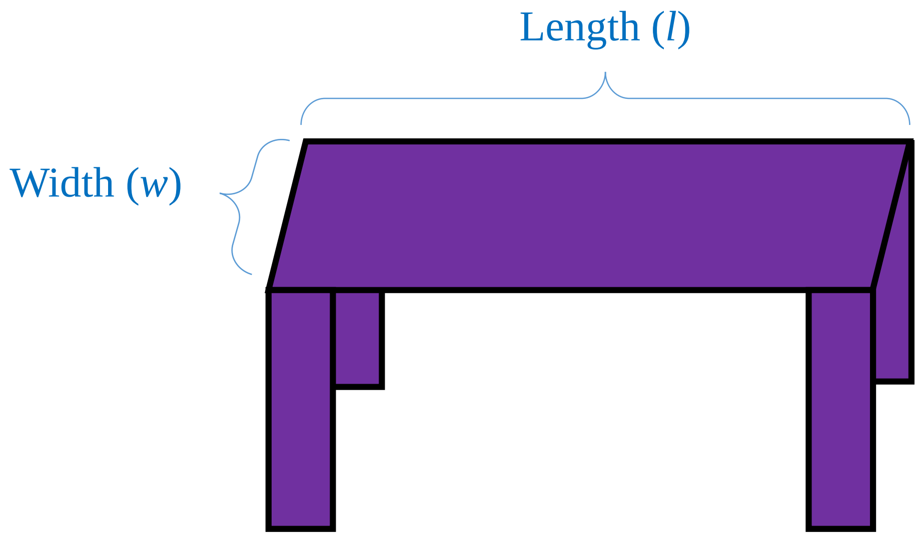
Minimize:

$$f(l, w) = 1 * (l + w)$$

Suject to:

$$\frac{l}{w} = 1$$





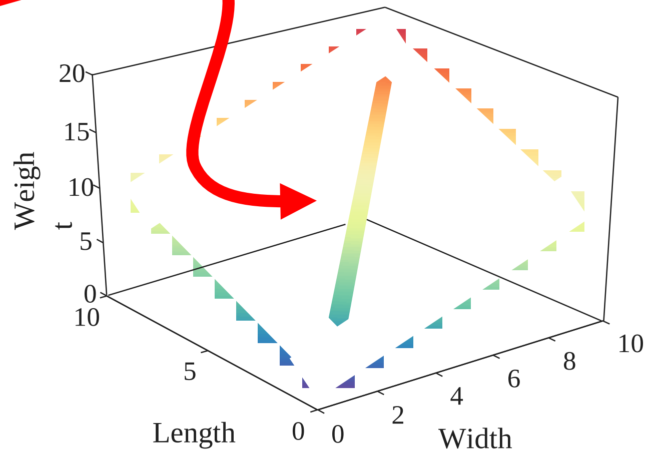
Constraints in designing a table

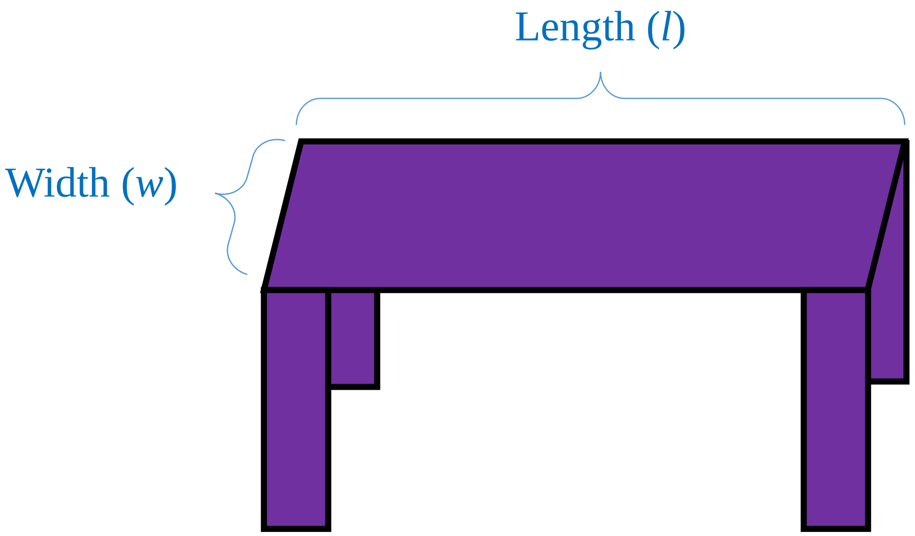
$Minimize$:

$$f(l, w) = 1 * (l + w)$$

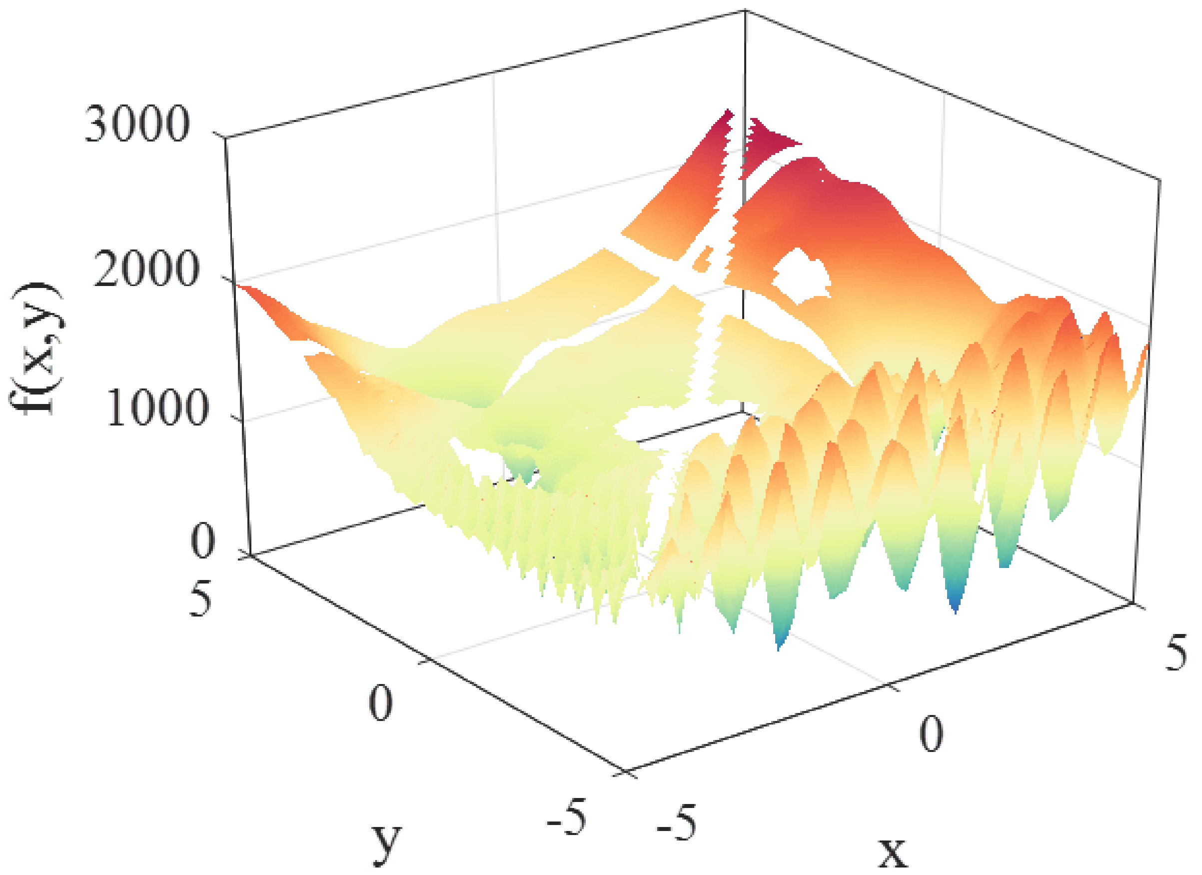
$Suject\ to$:

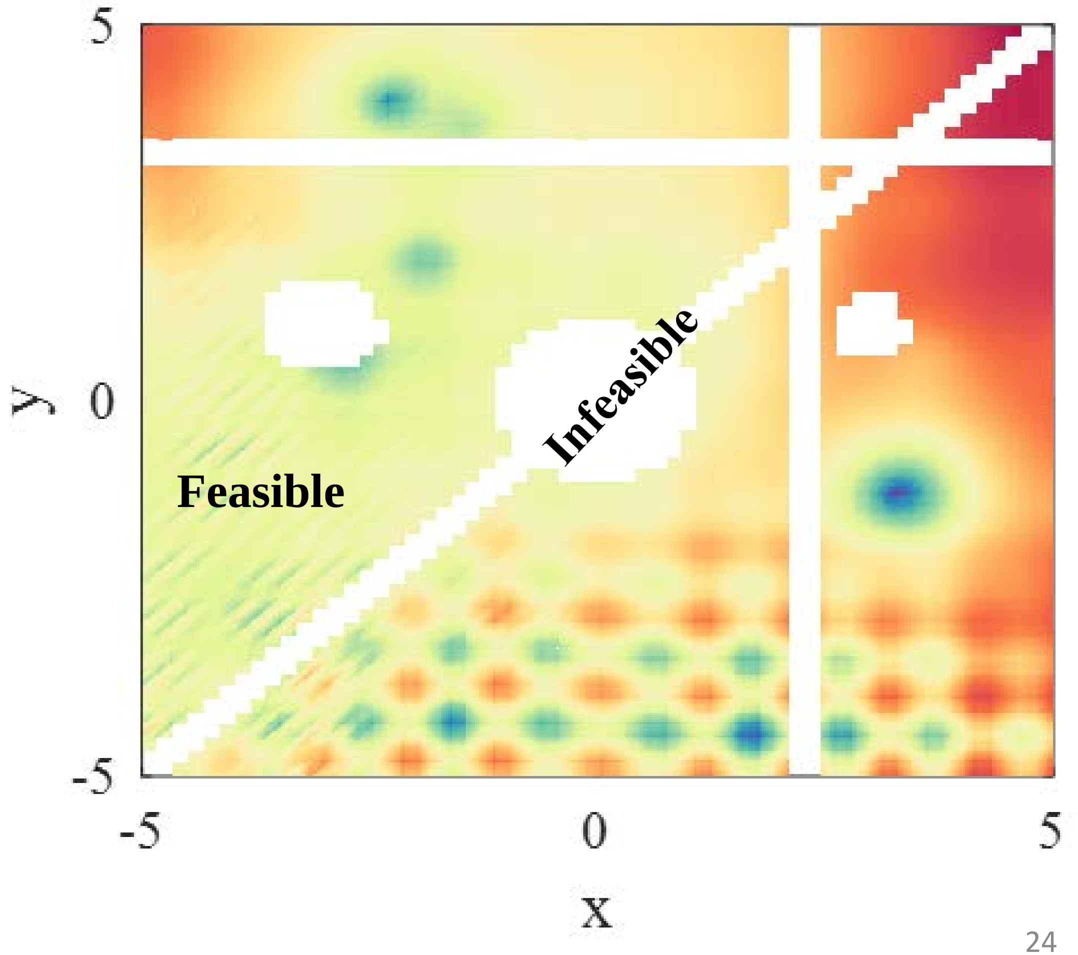
$$\frac{l}{w} = 1$$



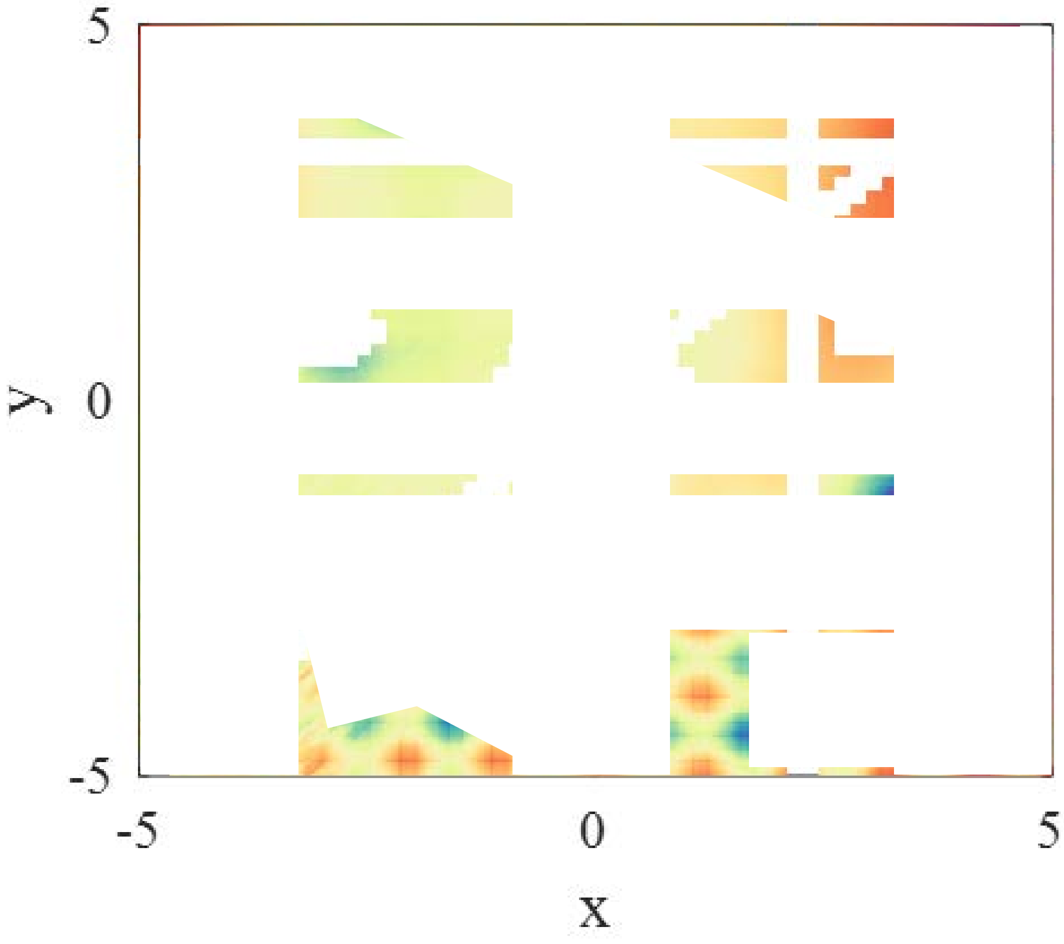


Constraints in a landscape

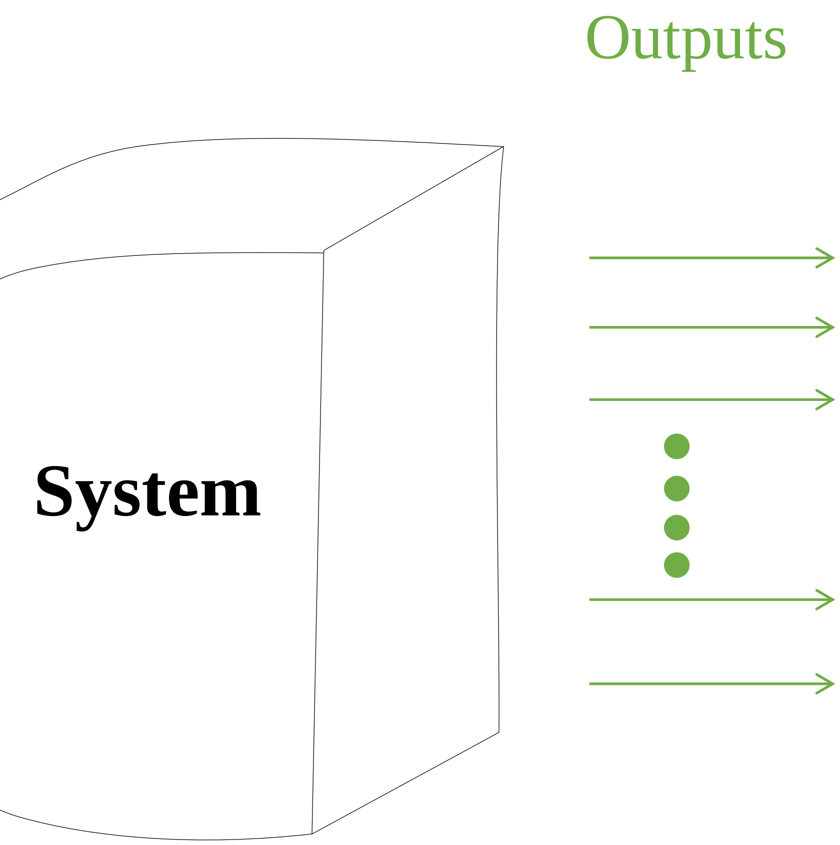




Highly constrained landscapes

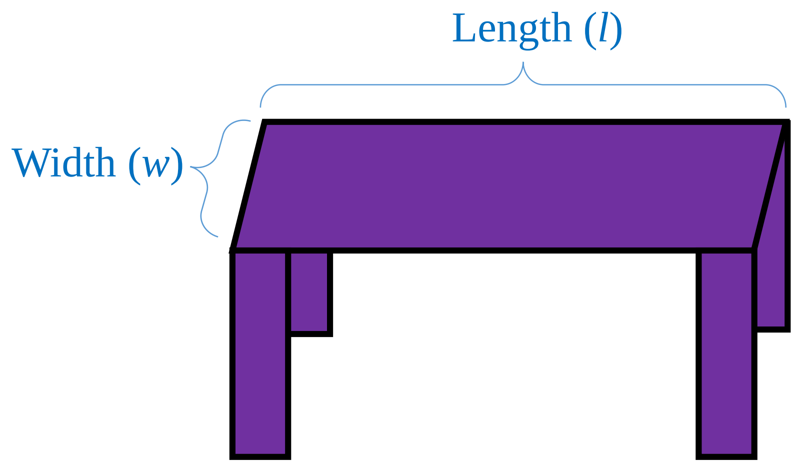


Objectives



- Multiple objectives
- Conflicting objectives
- Dynamic objectives
- A large number of objectives
- …

Table design example



$$Minimize:$$

$$\textbf{Weight}: \quad f_1(l, w) = 1 * (l + w)$$

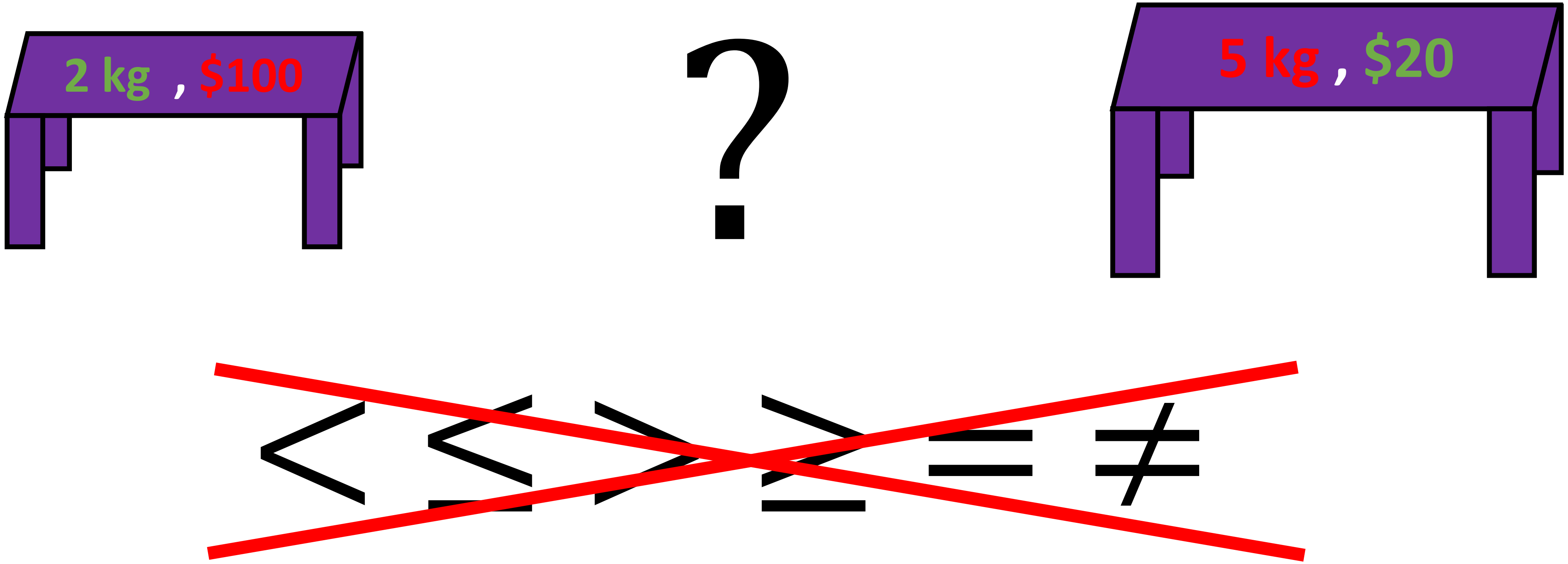
$$\textbf{Price}: \quad f_2(l, w) = l * w$$

$$Suject\ to:$$

$$\frac{l}{w} = 1$$

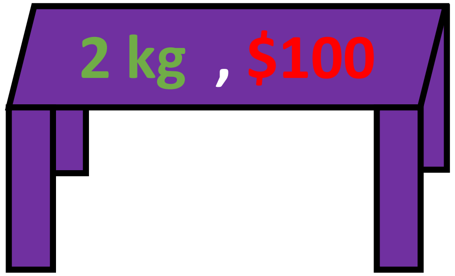
Comparing tables with two objectives

$\boldsymbol{Weight}$: $\boldsymbol{f_1(l, w)}$ **and** $\boldsymbol{Price}$: $\boldsymbol{f_2(l, w)}$



Comparing tables with two objectives

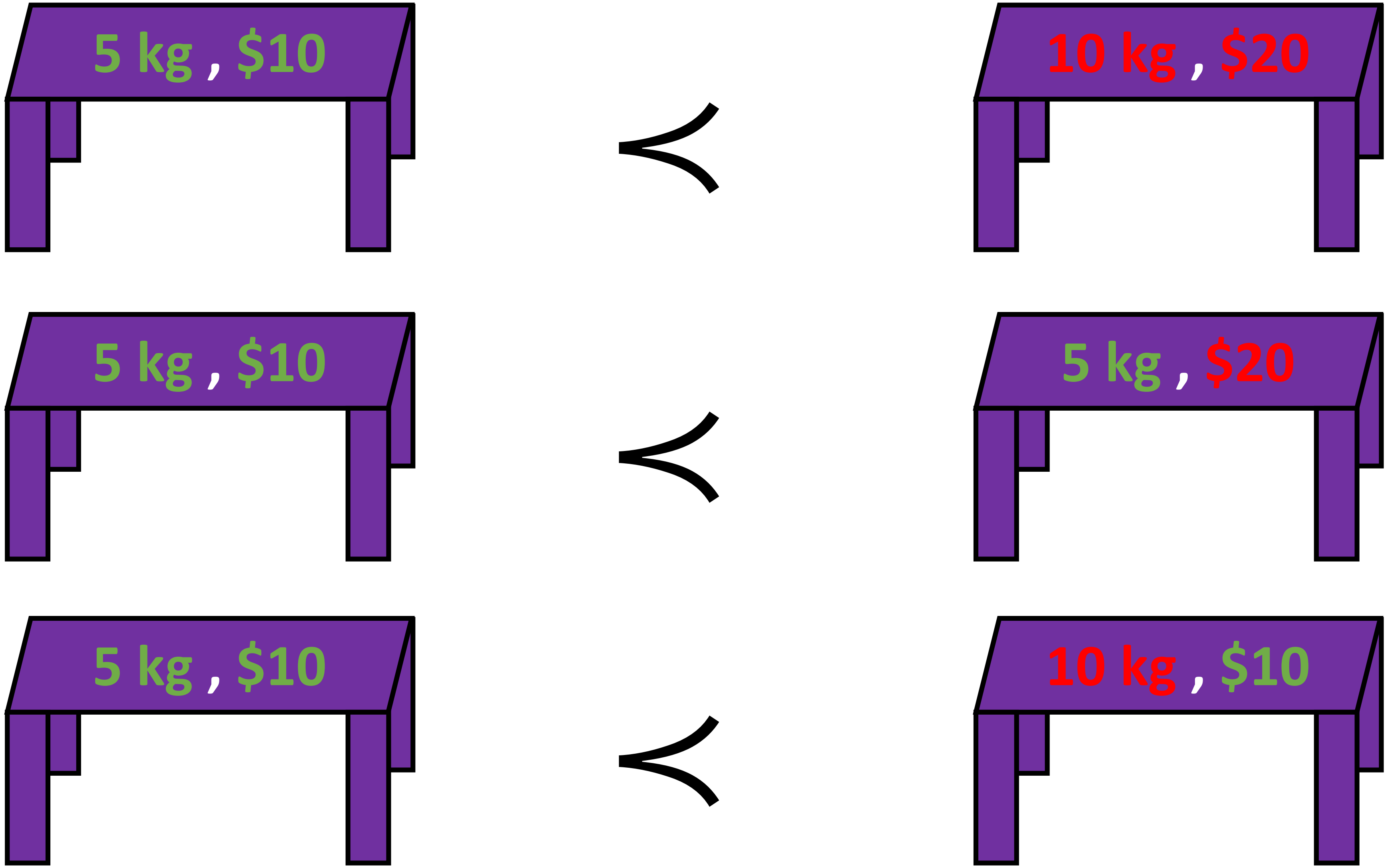
$\boldsymbol{Weight}$: $\boldsymbol{f_1(l, w)}$ and $\boldsymbol{Price}$: $\boldsymbol{f_2(l, w)}$



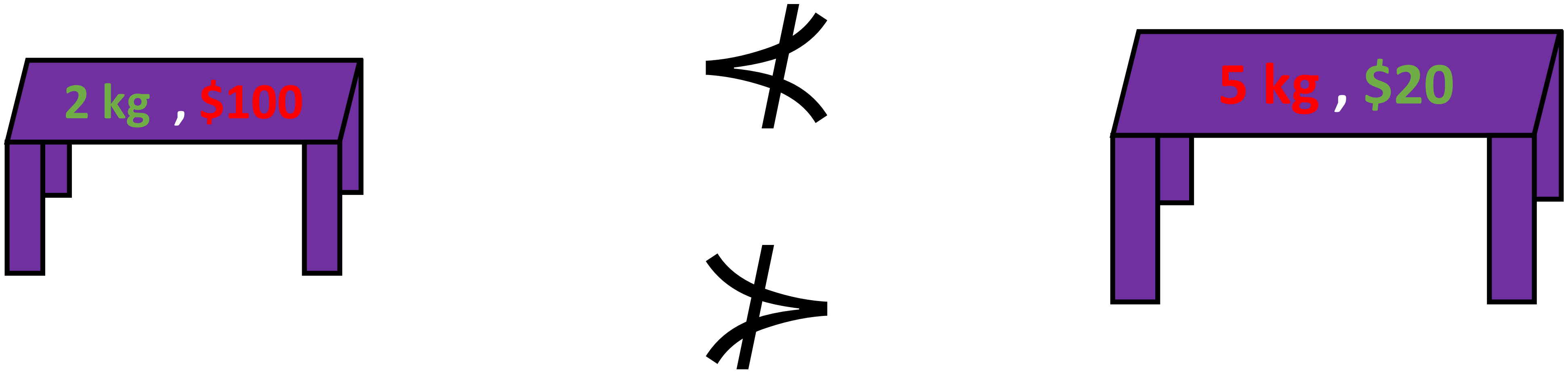
?

$\prec$

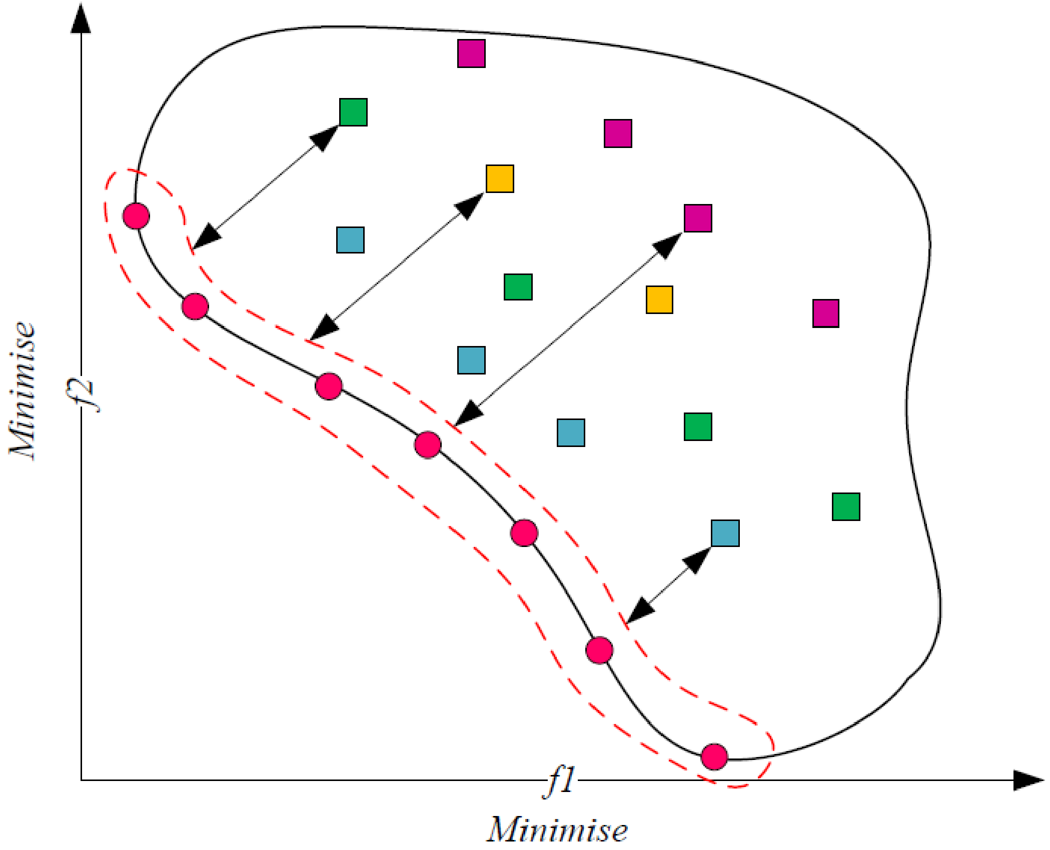
Pareto optimal dominance



Pareto optimal solutions



Pareto optimal solutions



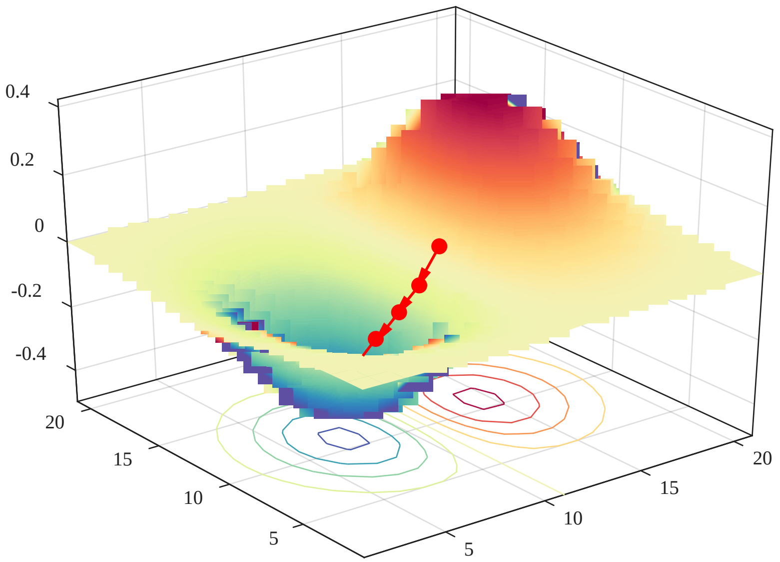
OPTIMIZATION ALGORITHMS

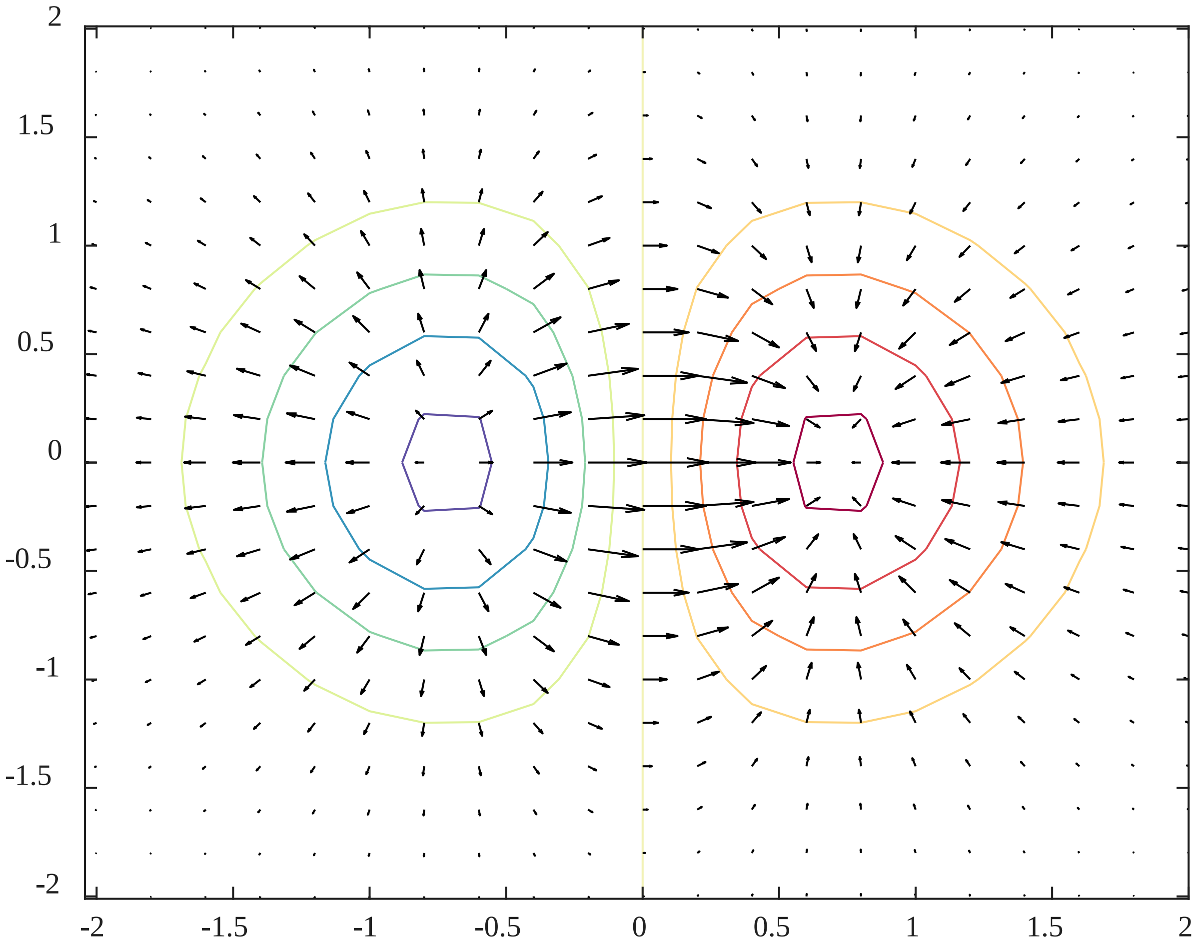
Conventional vs. Modern

Deterministic vs. Stochastic

Conventional (deterministic) optimization algorithms

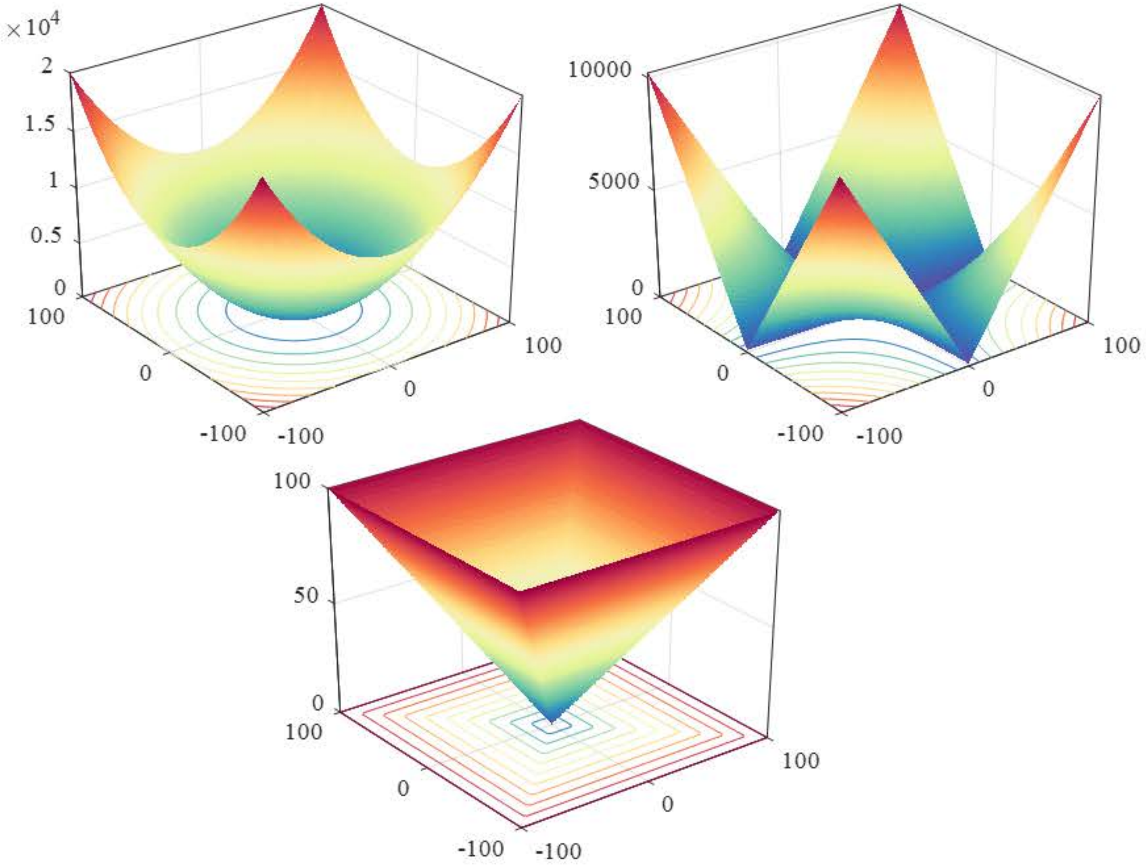
- Gradient descent algorithm





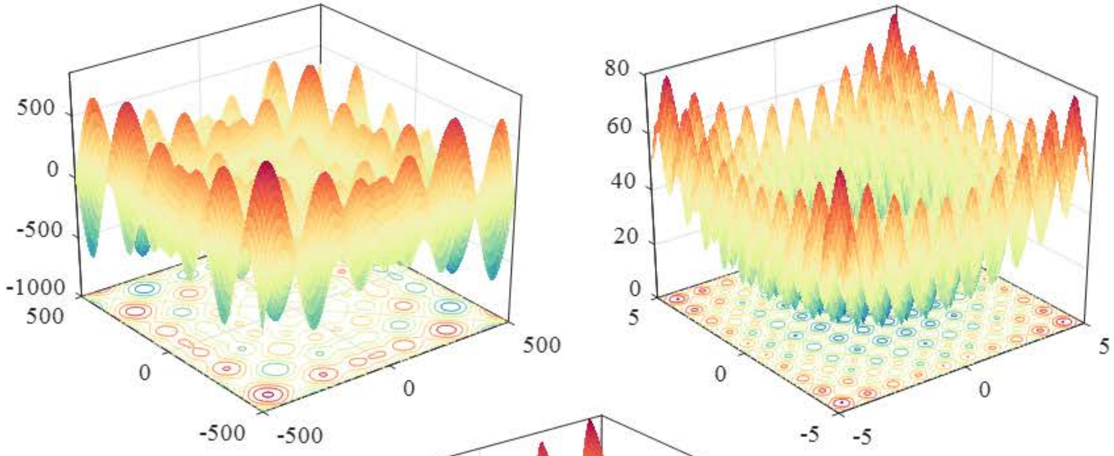
Gradient descent algorithm

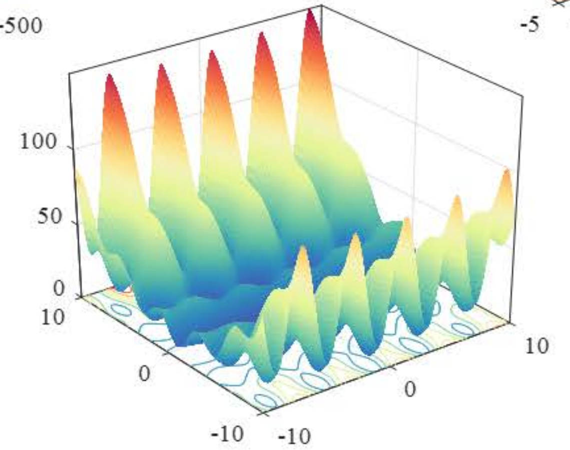
Efficient for unimodal landscapes



Not efficient for unimodal landscapes

Highly depends on the starting point





Modern (stochastic) optimization algorithms 

- Modern algorithms

It gives different outputs

There are random components

Modern (stochastic) optimization algorithms 

Deterministic

Advantages:

- Reliable in finding the same solution
- Require less number of function evaluation
- Fast convergence

Drawbacks:

- Local optima stagnation
- Low chance of finding the global optimum
- High dependency on the initial solution
- Mostly need gradient

Stochastics

Advantages:

- Avoid local solutions
- Higher chance of finding the global optimum
- Low dependency on the initial solution
- Mostly do not need gradient

Drawbacks:

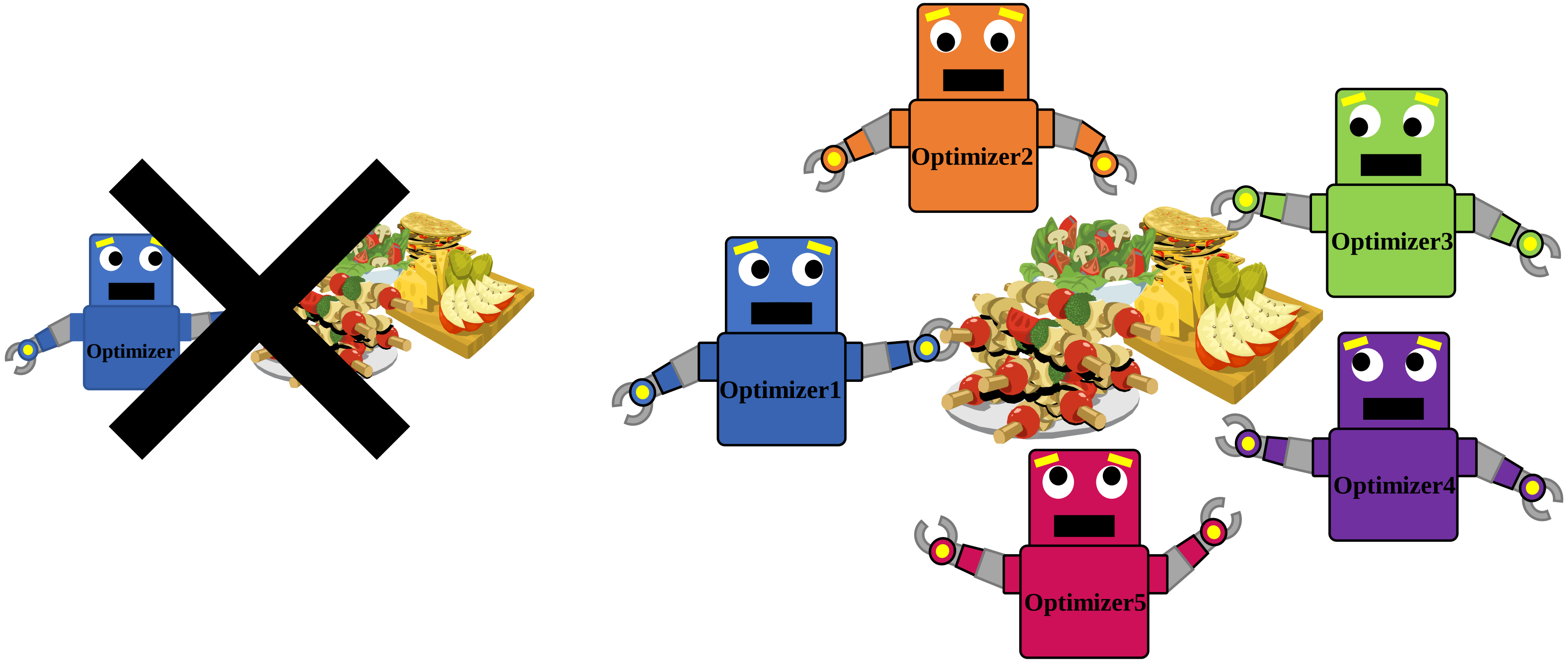
- Slow convergence speed
- Finding different answers in each run

Modern (stochastic) optimization algorithms

High local optima avoidance

Gradient-free mechanism

No Free Lunch (NFL) theorem



There is **no optimization algorithm** to solve **all optimization problems**

Wolpert, David H., and William G. Macready. "No free lunch theorems for optimization." IEEE transactions on evolutionary computation 1.1 (1997): 67-82.

Your role

PROBLEM

?

ALGORITHMS

Your role

PROBLEM

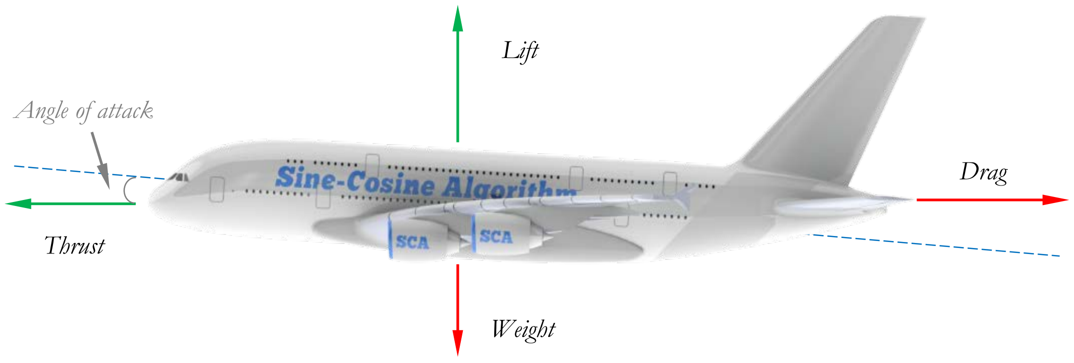
ALGORITHMS

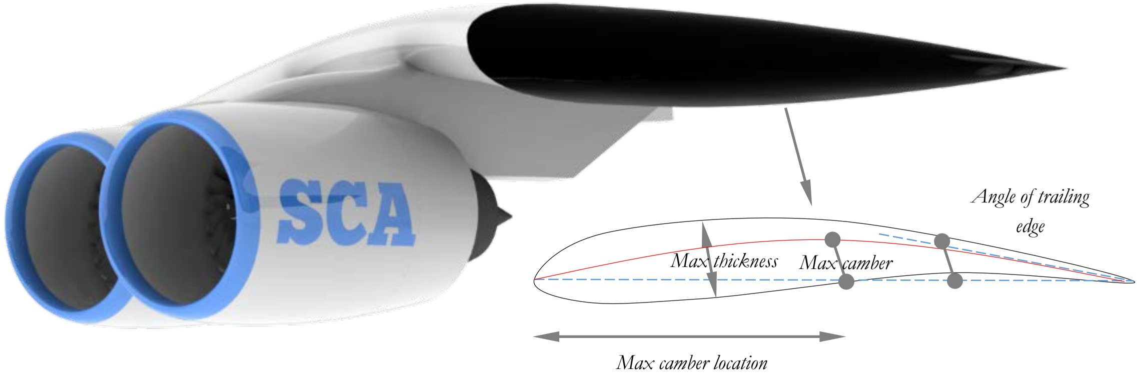
Technical challenges when using commercial simulators

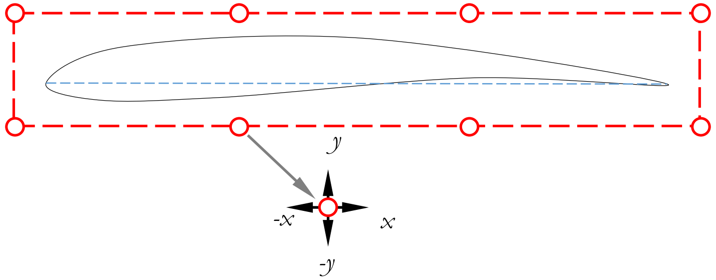
- Most of simulators have simple optimization toolboxes.
- We need to employ better recent optimization algorithms.
- There are many issues in connecting MATLAB to the simulator.
- Since that optimization requires a large number of simulations, it is necessary to run a simulator in parallel.

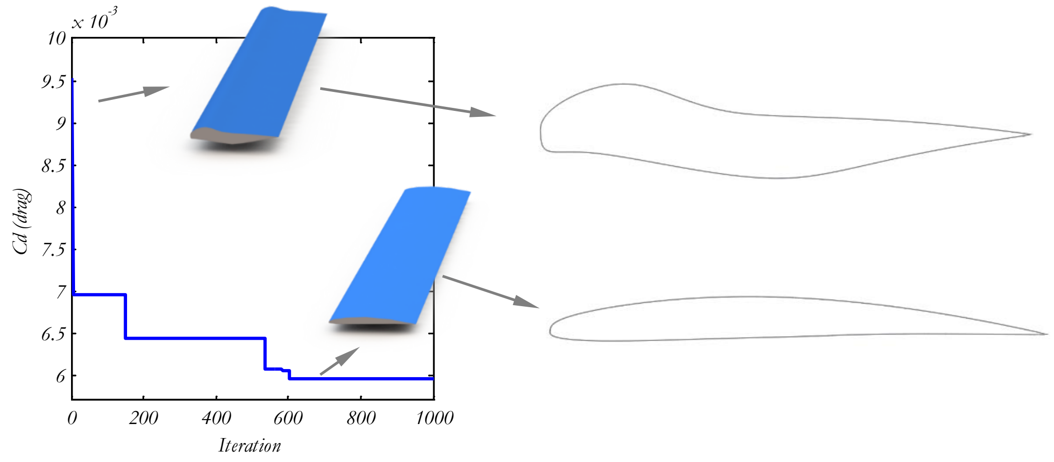
Thank You

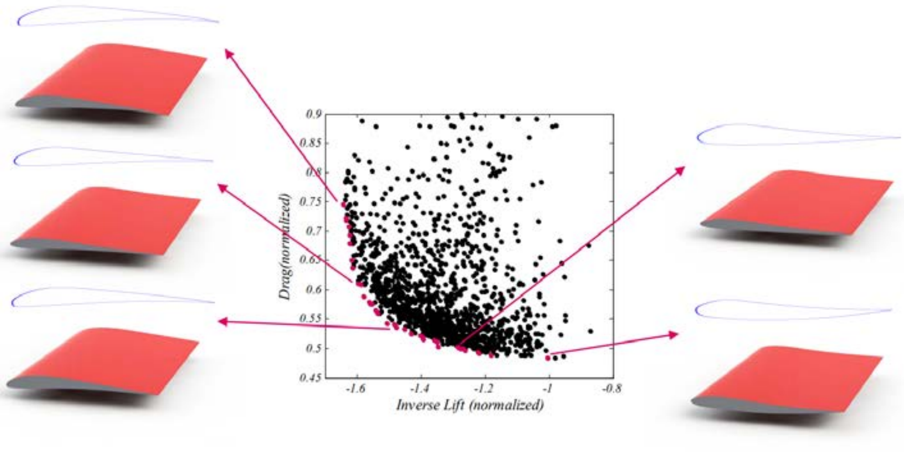
Case study 1


Lift
Angle of attack
Sine-Cosine Algorithm
SCA
SCA
Drag
Thrust
Weight

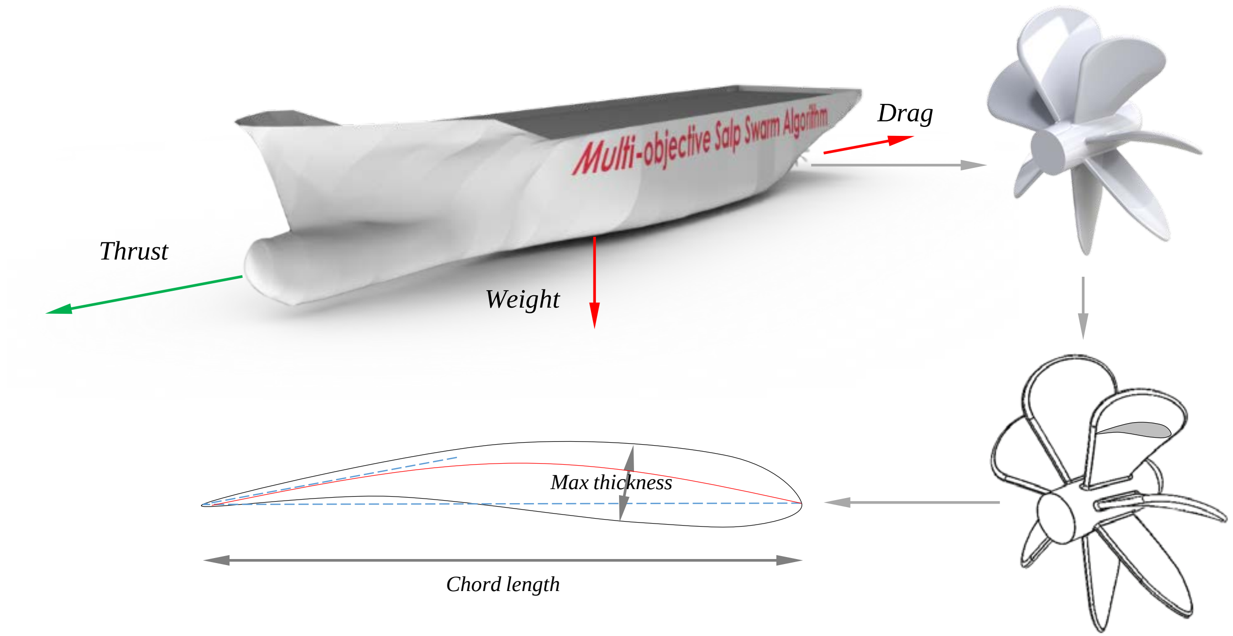

SCA
Angle of trailing edge
Max thickness
Max camber
Max camber location

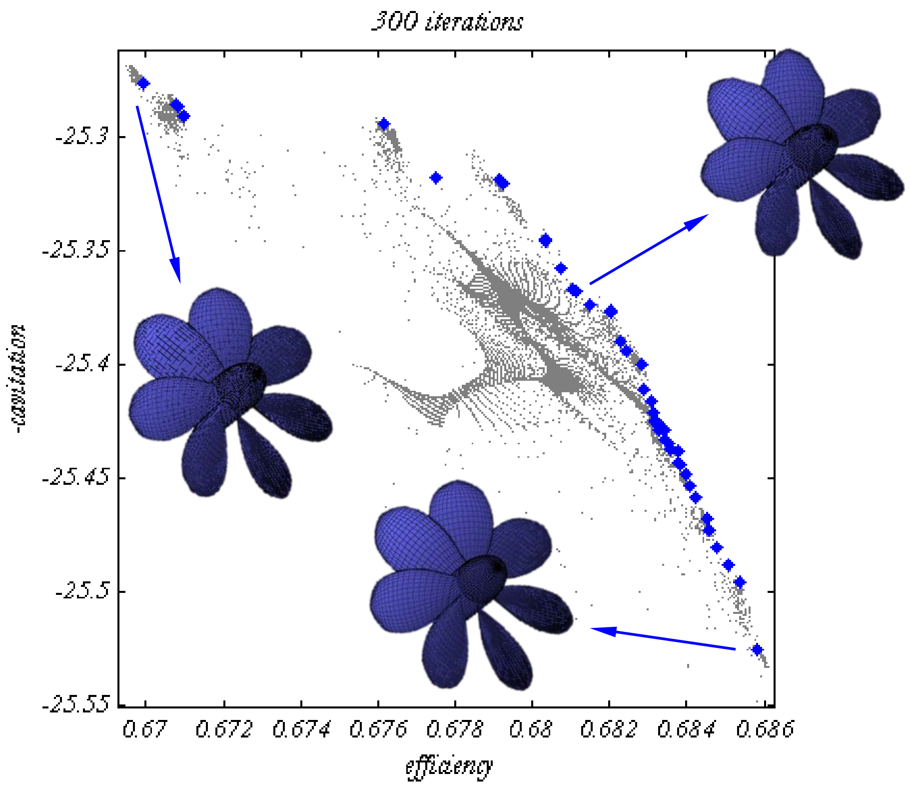

y
-x
x
-y


x 10^-3
10
9.5
9
8.5
8
7.5
7
6.5
6
Cd (drag)
0
200
400
600
800
1000
Iteration


Drag(normalized)
Inverse Lift (normalized)

Case study 2


Multi-objective Salp Swarm Algorithm
Drag
Thrust
Weight
Max thickness
Chord length


300 iterations
-25.3
-25.35
-25.4
-25.45
-25.5
-25.55
0.67
0.672
0.674
0.676
0.678
0.68
0.682
0.684
0.686
efficiency
-cavitation