Introduction to Optimization

Sergio Peignier

Why Optimization?

Core Idea

In many real-world problems, we want to:

find the best possible decision

- What is the **shortest route** between cities?
- How to train a model with **minimum error**?
- How to allocate resources **efficiently**?
- How to design systems with **best performance**?

Key Question

How can we systematically find the best solution?

Motivating Example: Finding the Best Route

Problem

You want to visit several cities:

Lyon, Madrid, Mexico, Buenos Aires, La Paz

- Many possible routes:
  - Lyon → Buenos Aires → La Paz → Madrid → Mexico
  - Madrid → Mexico → Lyon → Buenos Aires → La Paz
  - Lyon → Buenos Aires → Madrid → Mexico → La Paz
- Each route has a different total distance
- Goal: **find the shortest one**

Insight

- Each route = a **candidate solution**
- Total distance = **objective function**
- We want to **minimize** it

Optimization Problem

Definition

Let $f : A \to \mathbb{R}$. A constrained optimization problem:

$$\begin{aligned} \min_{a \in A} \quad & f(a) \\ \text{s.t.} \quad & g(a) \leq 0, \quad h(a) = 0 \end{aligned}$$

- A: search space
- a: candidate (feasible) solution
- $f(a)$: objective (cost / loss / utility)
- $g(a) \leq 0$ and $h(a) = 0$: constrains

Minimizers and Notation

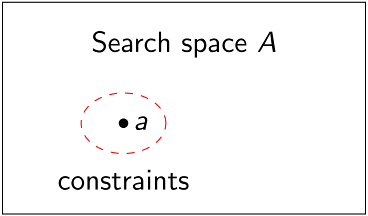
$$\text{ArgMin}_{a \in A} f(a)$$

- Set of optimal solutions
- If unique: $a^* = \text{ArgMin} f(a)$
- Minimizer: $f(a^*) \leq f(a)$

Remark

Maximization : minimize $(-f)$

Search Space and Constraints



- Only feasible solutions are allowed
- Constraints define the search region

Applications

- **Machine Learning**: minimize prediction error
- **Statistics**: maximize likelihood
- **Logistics**: minimize delivery distance
- **Engineering**: optimize energy consumption

Example

Train a model by minimizing:

$$\text{error}(w) = \|Xw - y\|^2$$

Discrete vs Continuous

Continuous

$$A \subseteq \mathbb{R}^n$$

- Regression coefficients
- Finding a barycenter

Discrete

- Traveling Salesman Problem (TSP)
- Knapsack problem

Discrete Example: TSP

- Cities: Lyon, Madrid, Mexico, Buenos Aires, La Paz
- Goal: minimize total distance

$$(\text{Lyon}, \text{Madrid}, \text{Mexico}, \ldots)$$

Challenge

Number of solutions grows factorially:

$$(n-1)!/2$$

Local vs Global Minima

Local Minimum

Best solution a^* in a neighborhood

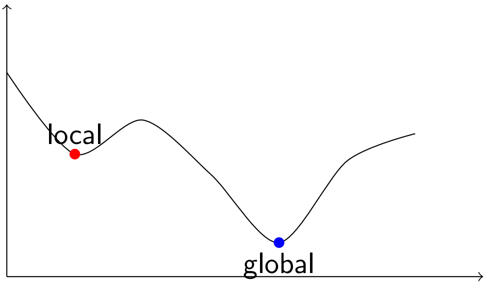
$$\exists\, \delta > 0 \text{ s.t. } \forall a \in A,\ \|a - a^*\| \leq \delta \ \Rightarrow\ f(a') \leq f(a)$$

Global Minimum

Best solution a^* overall

$$\forall a \in A,\ f(a^*) \leq f(a)$$

Illustration



Convexity

Definition

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

- No “valleys inside valleys”
- Shape like a bowl

Key Theorem

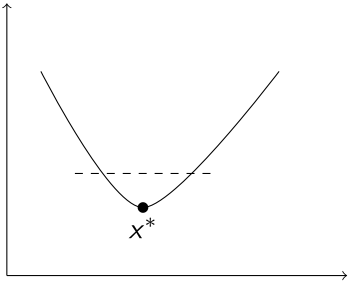
Theorem

If f is convex:

$$\text{local minimum} = \text{global minimum}$$

- Very important in optimization
- Guarantees correctness of algorithms

Convexity Intuition



- Tangent line lies below the function
- No better solution exists elsewhere

Multiobjective Optimization

$$\min(f_1(a), f_2(a), \ldots, f_k(a))$$

- Several objectives
- Often conflicting

Pareto Optimality

Dominance

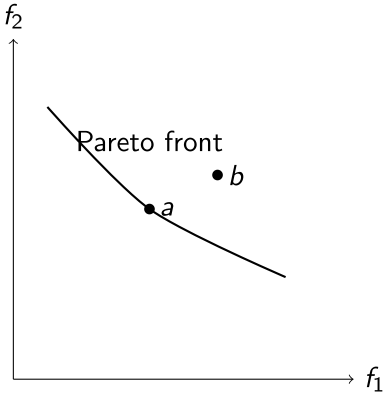
a dominates b if:

$$f_i(a) \leq f_i(b) \; \forall i$$

Pareto Optimal

No other solution is better in all objectives

Pareto Front



- *a* is optimal
- *b* is dominated

Examples

- Car design:
  - minimize fuel consumption
  - maximize speed
- Smartphone:
  - maximize battery life
  - minimize weight
- Machine Learning:
  - minimize error
  - minimize computation cost