## Plan for Today

- Evolutionary Algorithms
  - Optimization Basics
  - EA Basics

---

### Optimization Basics

#### given:

- **objective function** $f : X \rightarrow \mathbb{R}$
- **feasible region** $X$ (= nonempty set)

**objective:** find solution with *minimal or maximal* value!

#### optimization problem:

find $x^* \in X$ such that $f(x^*) = \min \{ f(x) : x \in X \}$

**note:**

$max \{ f(x) : x \in X \} = -\min \{ -f(x) : x \in X \}$

---

#### modelling

- input

#### simulation

- system

#### optimization

- output
Optimization Basics

Lecture 10

local solution \( x^* \in X \):

\[ \forall x \in N(x^*) : f(x^*) \leq f(x) \]

if \( x^* \) local solution then

f(x^*) local optimum / minimum

neighborhood of \( x^* \) = bounded subset of \( X \)

example: \( X = \mathbb{R}^n, N(x^*) = \{ x \in X : \| x - x^* \|_2 \leq \varepsilon \} \)

remark:

evidently, every global solution / optimum is also local solution / optimum;

the reverse is wrong in general!

\[ f(x) = a_1 x_1 + \ldots + a_n x_n \rightarrow \max! \]

with \( x_i \in \{0,1\}, a_i \in \mathbb{R} \)

add constant \( g(x) = b_1 x_1 + \ldots + b_n x_n \leq b \)

\( \Rightarrow \) \( x^*_i = 1 \) if \( a_i > 0 \)

\( \Rightarrow \) NP-hard

add capacity constraint to TSP \( \Rightarrow \) CVRP

\( \Rightarrow \) still harder

What makes optimization difficult?

some causes:

- local optima (is it a global optimum or not?)
- constraints (ill-shaped feasible region)
- non-smoothness (weak causality) \( \Rightarrow \) strong causality needed!
- discontinuities (\( \Rightarrow \) nondifferentiability, no gradients)
- lack of knowledge about problem (\( \Rightarrow \) black / gray box optimization)

When using which optimization method?

mathematical algorithms

- problem explicitly specified
- problem-specific solver available
- problem well understood
- resources for designing algorithm affordable
- solution with proven quality required

⇒ \textbf{don't apply EAs}

randomized search heuristics

- problem given by black / gray box
- no problem-specific solver available
- problem poorly understood
- insufficient resources for designing algorithm
- solution with satisfactory quality sufficient

⇒ \textbf{EAs worth a try}

Evolutionary Algorithm Basics

Idea:

using biological evolution as metaphor and as pool of inspiration

⇒ interpretation of biological evolution as iterative method of improvement

feasible solution \( x \in X = S_1 \times \ldots \times S_n \) = chromosome of individual

multiset of feasible solutions = population: multiset of individuals

objective function \( f : X \rightarrow \mathbb{R} \) = fitness function

Often: \( X = \mathbb{R}^n, X = [0,1]^n, X = \mathbb{P}_n = \{ \pi : \pi \text{ is permutation of } \{1,2,\ldots,n\} \} \)

Also: combinations like \( X = \mathbb{R}^n \times \mathbb{B}^p \times \mathbb{P}_q \) or non-cartesian sets

⇒ structure of feasible region / search space defines representation of individual
Evolutionary Algorithm Basics

Lecture 10

Evolutionary Algorithm Basics

Lecture 10

Evolutionary Algorithm Basics

Lecture 10

Evolutionary Algorithm Basics

Lecture 10

Algorithmic skeleton

- Initialize population
- Evaluation
- Parent selection
- Variation (yields offspring)
- Evaluation (of offspring)
- Survival selection (yields new population)
- Stop?

Output: best individual found

Specific example: (1+1)-EA in \( \mathbb{R}^n \) for minimizing some \( f: \mathbb{R}^n \to \mathbb{R} \)

- Population size = 1, number of offspring = 1, selects best from 1+1 individuals
- No choice, here

1. Initialize \( X^{(0)} \in \mathbb{R}^n \) uniformly at random, set \( t = 0 \)
2. Evaluate \( f(X^{(0)}) \)
3. Select parent: \( Y = X^{(t)} \)
4. Variation: Flip each bit of \( Y \) independently with probability \( p_m = 1/n \)
5. Evaluate \( f(Y) \)
6. Selection: If \( f(Y) \leq f(X^{(0)}) \) then \( X^{(t+1)} = Y \) else \( X^{(t+1)} = X^{(t)} \)
7. If not stopping then \( t = t + 1 \), continue at (3)

Selection

- Select parents that generate offspring \( \rightarrow \) selection for reproduction
- Select individuals that proceed to next generation \( \rightarrow \) selection for survival

Necessary requirements:
- Selection steps must not favor worse individuals
- One selection step may be neutral (e.g., select uniformly at random)
- At least one selection step must favor better individuals

Typically: Selection only based on fitness values \( f(x) \) of individuals
Seldom: Additionally based on individuals’ chromosomes \( x \) \( \rightarrow \) maintain diversity

Selection methods

- Population \( P = (x_1, x_2, ..., x_\mu) \) with \( \mu \) individuals
- Two approaches:
  1. Repeatedly select individuals from population with replacement
  2. Rank individuals somehow and choose those with best ranks (no replacement)

  - Uniform/neutral selection
    choose index \( i \) with probability \( 1/\mu \)
  - Fitness-proportional selection
    choose index \( i \) with probability \( s_i = \frac{f(x_i)}{\sum_{x \in P} f(x)} \)

Problems: \( f(x) > 0 \) for all \( x \in X \) required \( \Rightarrow g(x) = \exp(f(x)) > 0 \)
But already sensitive to additive shifts \( g(x) = f(x) + c \)
Almost deterministic if large differences, almost uniform if small differences

Don’t use!
Evolutionary Algorithm Basics

Lecture 10

Selection methods
population \( P = (x_1, x_2, ..., x_\mu) \) with \( \mu \) individuals

- **rank-proportional selection**
  order individuals according to their fitness values
  assign ranks
  fitness-proportional selection based on ranks
  \( \Rightarrow \) avoids all problems of fitness-proportional selection
  but: best individual has only small selection advantage (can be lost!)

- **k-ary tournament selection**
  draw \( k \) individuals uniformly at random (typically with replacement) from \( P \)
  choose individual with best fitness (break ties at random)
  \( \Rightarrow \) has all advantages of rank-based selection and
  probability that best individual does not survive:
  \[ \left(1 - \frac{1}{\mu}\right)^{k\mu} \approx e^{-k} \]

Selection methods without replacement
population \( P = (x_1, x_2, ..., x_\mu) \) with \( \mu \) parents and
population \( Q = (y_1, y_2, ..., y_\lambda) \) with \( \lambda \) offspring

- \((\mu, \lambda)\)-selection or truncation selection on offspring or comma-selection
  rank \( \lambda \) offspring according to their fitness
  select \( \mu \) offspring with best ranks
  \( \Rightarrow \) best individual may get lost, \( \lambda \geq \mu \) required

- \((\mu+\lambda)\)-selection or truncation selection on parents + offspring or plus-selection
  merge \( \lambda \) offspring and \( \mu \) parents
  rank them according to their fitness
  select \( \mu \) individuals with best ranks
  \( \Rightarrow \) best individual survives for sure