Artificial Intelligence

Evolutionary Algorithms

Lesson 8: Multi-Criteria Optimization

Vincenzo Piuri

Università degli Studi di Milano

Contents

- Multi-criteria optimization
- Pareto-optimal solutions
- Pareto-optimal solutions with Evolutionary Algorithms
- Preventing the genetic drift

Multi-Criteria Optimization (1)

- Optimization of more than one variable
  - Achieve different objectives as good as possible
  - Example: buying a car
    - Low price
    - Low fuel consumption
    - Best comfort
- Different objectives may be conflicting
  - Example:
    - Additional charge for most comfort
    - Air conditioning or large inner space require a big engine: higher pricing, more fuel consumption

Multi-Criteria Optimization (2)

- k criteria given, with one objective function each

$$f_i : \Omega \to \mathbb{R}, \qquad i = 1, \dots, k$$

- **Simple** approach
  - Combine the k objective functions into one aggregated objective function

$$f(s) = \sum_{i=1}^{k} w_i \cdot f_i(s)$$

Multi-Criteria Optimization (3)

- Problems of simple approach
  - Relative significance of criteria needs to be fixed before
  - Choice of weights not always simple
- In general: problem of **aggregation of preference**
  - Arrow's impossibility theorem: there is no choice function that has all the desired features
  - Scaled order of preferences
    - Finding a suitable scaling is more complex than finding suitable weights

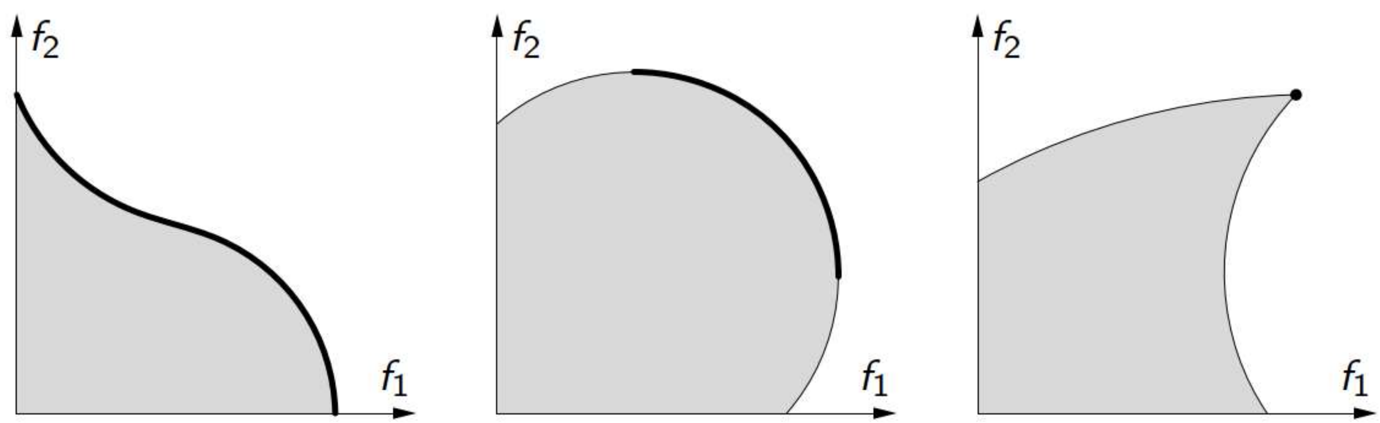
Pareto-Optimal Solutions (1)

- An element $s \in \Omega$ is called **Pareto-optimal** regarding the objective functions f_i , $i = 1,\ldots,k$, if there is no such element $s' \in \Omega$ for which

$$\begin{array}{lll} \forall i, 1 \leq i \leq k: & f_i(s') \geq f_i(s) & \text{and} \\ \exists i, 1 \leq i \leq k: & f_i(s') > f_i(s) & \text{holds.} \end{array}$$

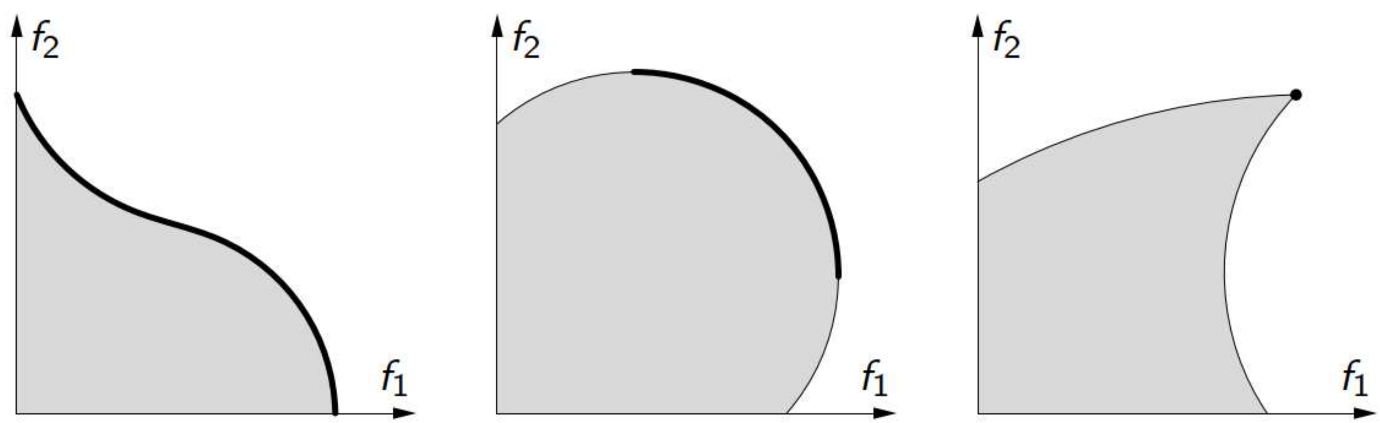
Pareto-Optimal Solutions (2)

- Pareto-optimal
  - No value of an objective function can get better without the value of another function getting worse
- The set of Pareto-optimal elements is called **Pareto-front**



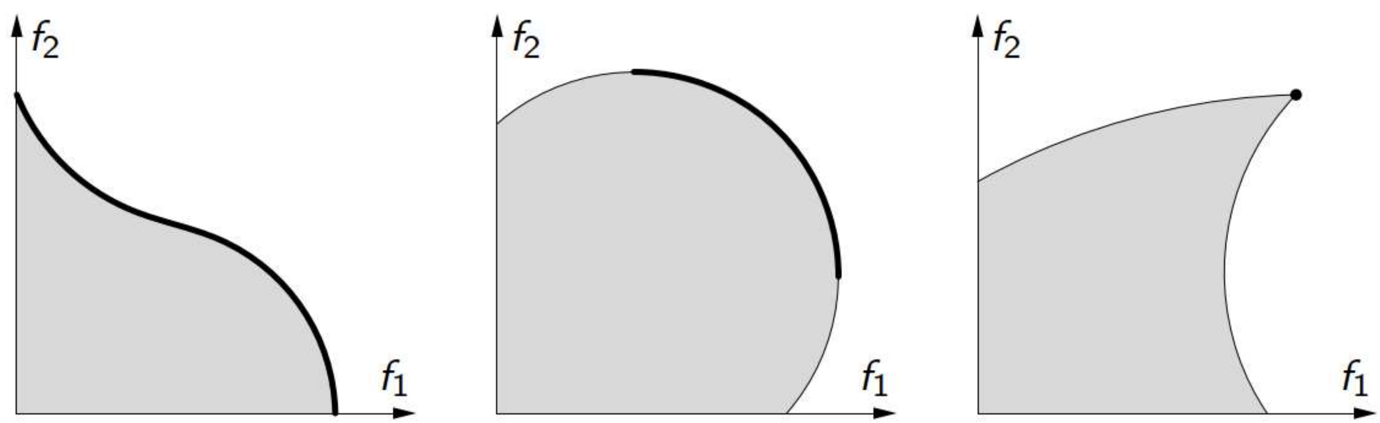
Pareto-Optimal Solutions (3)

- Advantages when searching for Pareto-optimal solutions
  - No need for aggregating objective functions
  - No need for choosing weights
  - The search has to be performed only once
  - After this search the solutions are chosen



Pareto-Optimal Solutions (4)

- All points of Ω are located within the gray zone
- Pareto-optimal solution = bold part of the border
  - Pareto-optimal solution can be unique (depending on the location of the candidate solutions)



Pareto-Optimal Solutions with EAs (1)

- Objective
  - Spreading the population along the Pareto-Front as widely as possible
- Challenge
  - Without previously defined weights there are many different, equivalent solutions
- Simplest approach
  - Weighted sum of the objective functions as fitness function

Pareto-Optimal Solutions with EAs (2)

- **Vector Evaluated Genetic Algorithm**
  - given k criteria with assigned objective functions $f_i, i = 1, \ldots, k$
  - $\forall\, i, 1, \ldots, k$: choose $|P|/k$ individuals according to the fitness function f_i
- Advantage
  - simple, without much computational effort
- Disadvantage
  - clear handicap for solutions that satisfy every criterion good, but none perfectly
- Consequence
  - search concentrates on marginal solutions

Pareto-Optimal Solutions with EAs (3)

- Selection by dominance
  - Rating scale of the individuals of one population
  - Find all non-dominated solutions of a population
  - Assessing solution candidates to the best rank, remove them from the population
  - Repeat identification and removal of non-dominated solution candidates for other ranks, until population is empty

Pareto-Optimal Solutions with EAs (4)

- Perform a rank-based selection according to the ranking scale
- Problem
  - All individuals of the Pareto-Front are assessed as equally good
  - **Genetic drift**: pareto-Front converges at a random point, because of random effects

Preventing the Genetic Drift

- Objective
  - Spread along the Pareto-front as equally as possible
- Solution
  - Niche techniques to be able to decide between individuals with same rank
  - E.g. **power law sharing**
    - If more individuals have similar fitness score, the new individual similar to them will have lower fitness score
- Problem
  - Calculating the ranking scale is costly

Non-Dominated Sorted Genetic Algorithm (1)

1. Generate offsprings by applying genetic operators to the parent population
2. Select non-dominated individuals from parents+offsprings population: they constitutes the front F_i at the i-th iteration
3. Remove individuals in F_i from current population
4. Repeat steps 2 and 3 until the current population is empty
5. Fill the new generation by taking the individuals from the fronts F_i, from i=1 forward, until the complete F_i can be included
6. Apply crowding by similarity to the next front and extract the missing individuals to complete the new generation

Non-Dominated Sorted Genetic Algorithm (2)

- Poor approximation of the Pareto-Front
  - Parameter setting of niche radius ε
  - Population used for two purposes
    - As storage for non-dominated individuals (Pareto-Front)
    - As living population (for searching the search space)
  - Archive may become big

Strength Pareto Evolutionary Algorithm

- Simple evolutionary algorithm
- Evaluation function in two components
  1. How many individuals dominate individuals dominating this individual
  2. Distance to the $\sqrt{n}$-th closest individual
- Addressing the limits of Non-Dominated Sorted Genetic Algorithm:
  – Separate archive for non-dominated individuals and for population
  – Test all individuals for dominance by archive individuals
  – Limit archive size: remove dominated individuals from the archive, includes new non-dominated individuals
  – If no enough room in archive: replace in archive, because of distance to other archived individuals
  – If new generation is incomplete: add fittest dominated individuals

Pareto-Archived Evolutionary Strategy

- (1 + 1)-evolution strategy (plus strategy with μ=1 and λ=1)
- Archive of non-dominated individuals
- Condition of acceptance of new individual:
  - Dominates an archived individual
  - No enough variance in population
- Niches
  - If no enough space in archive: remove individuals belonging to niches