Optimization and Simulation

Multi-objective optimization

Michel Bierlaire

Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne

Multi-objective optimization

Concept

- Need for minimizing several objective functions.
- In many practical applications, the objectives are conflicting.
- Improving one objective may deteriorate several others.

Examples

- Transportation: maximize level of service, minimize costs.
- Finance: maximize return, minimize risk.
- Survey: maximize information, minimize number of questions (burden).

Multi-objective optimization

$$\min_x F(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_P(x) \end{pmatrix}$$

subject to

$$x \in \mathfrak{F} \subseteq \mathbb{R}^n,$$

where

$$F : \mathbb{R}^n \to \mathbb{R}^P.$$

Why multi-objective optimization is different

Single-objective optimization

- Solutions can be totally ordered: better / worse.
- One optimal solution (or a small equivalent set).

Multi-objective optimization

- No total ordering between solutions.
- Many solutions are incomparable.
- The output is a set of solutions (Pareto set), not a single point.

Consequence

Optimization does not eliminate trade-offs — it reveals them.

Outline

Dominance

Dominance

Consider $x_1, x_2 \in \mathbb{R}^n$. x_1 is dominating x_2 if

1. x_1 is no worse in any objective

$$\forall i \in \{1, \ldots, p\}, f_i(x_1) \leqslant f_i(x_2),$$

2. x_1 is strictly better in at least one objective

$$\exists i \in \{1, \ldots, p\}, f_i(x_1) < f_i(x_2).$$

Notation

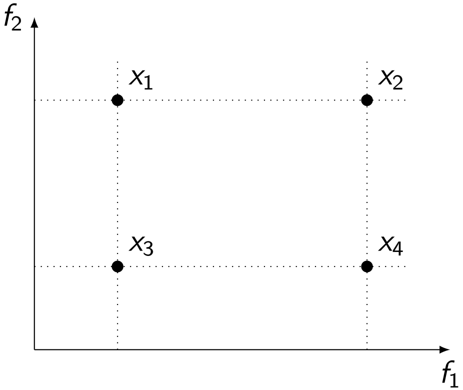
x_1 dominates x_2: $F(x_1) \prec F(x_2)$.

Dominance

Properties

- Not reflexive: $x \not\prec x$
- Not symmetric: $x \prec y \not\Rightarrow y \prec x$
- Instead: $x \prec y \Rightarrow y \not\prec x$
- Transitive: $x \prec y$ and $y \prec z \Rightarrow x \prec z$
- Not complete: $\exists x, y$: $x \not\prec y$ and $y \not\prec x$

Dominance: example



$$F(x_3) \prec F(x_2)$$
$$F(x_3) \prec F(x_1)$$
$$F(x_1) \nprec F(x_4)$$
$$F(x_4) \nprec F(x_1)$$

Optimality

Pareto optimality

The vector $x^* \in \mathcal{F}$ is Pareto optimal if it is not dominated by any feasible solution:

$$\nexists x \in \mathcal{F} \text{ such that } F(x) \prec F(x^*).$$

Intuition

x^* is Pareto optimal if no objective can be improved without degrading at least one of the others.

Optimality

Weak Pareto optimality

The vector $x^* \in \mathcal{F}$ is weakly Pareto optimal if there is no $x \in \mathcal{F}$ such that $\forall i = 1, \ldots, p,$

$$f_i(x) < f_i(x^*),$$

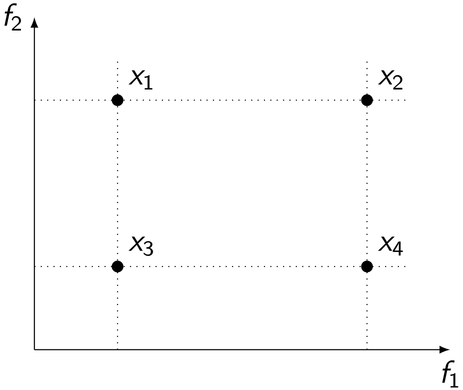
Intuition

Excludes solutions that are strictly worse in all objectives.

Pareto optimality

- P^*: set of Pareto optimal solutions
- WP^*: set of weakly Pareto optimal solutions
- $P^* \subseteq WP^* \subseteq \mathcal{F}$

Dominance: example



- x_3: Pareto optimal.
- x_1, x_3, x_4: weakly Pareto optimal.

Pareto frontier

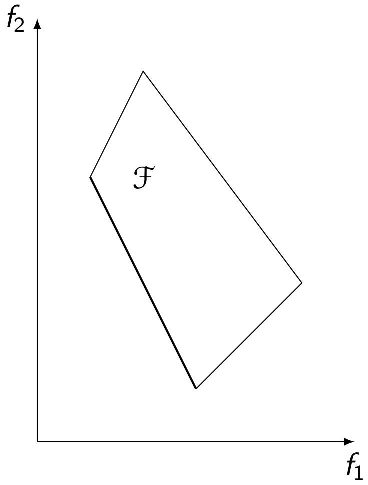
Pareto optimal set

$$P^* = \{x^* \in \mathfrak{F} | \nexists x \in \mathfrak{F} : F(x) \prec F(x^*)\}$$

Pareto frontier

$$PF^* = \{F(x^*) | x \in P^*\}$$

Pareto frontier



After optimization: the decision-maker

What optimization provides

- A set of Pareto optimal solutions.
- Explicit trade-offs between objectives.

What optimization does not provide

- A unique ‘best” solution.
- Preferences between objectives.

Key message

Choosing one solution requires preferences, policy goals, or external criteria.

Outline

Weighted sum

Weights

For each $i = 1, \ldots, p$, $w_i > 0$ is the weight of objective i.

Optimization

$$\min_{x \in \mathcal{F}} \sum_{i=1}^{p} w_i f_i(x). \tag{1}$$

Comments

- Weights may be difficult to interpret in practice.
- Generates a Pareto optimal solution.
- In the convex case, if x^* is Pareto optimal, there exists a set of weights such that x^* is the solution of (1).
- Non convex case: weighted sums cannot generate all Pareto optimal solutions.

Weighted sum: example

Train service

- f_1: minimize travel time
- f_2: minimize number of trains
- f_3: maximize number of passengers

Definition of the weights

- Transform each objective into monetary costs.
- Travel time: use value-of-time.
- Number of trains: estimate the cost of running a train.
- Number of passengers: estimate the revenues generated by the passengers.

Goal programming

Goals

For each $i = 1, \ldots, p$, g_i is the "ideal" or "target" objective function defined by the modeler.

Optimization

$$\min_{x \in \mathcal{F}} \|F(x) - g\|_\ell = \sqrt[\ell]{\sum_{i=1}^{p} |F_i(x) - g_i|^\ell}$$

Interpretation

- Emphasizes proximity to predefined targets.
- Encodes aspirations rather than trade-offs.
- Useful when objectives are constrained by policy or regulation.

Outline

Definitions

Transformations into single-objective

Lexicographic rules

Constrained optimization

Heuristics

Lexicographic optimization

Sorted objective

Assume that the objectives are sorted from the most important ($i = 1$) to the least important ($i = p$).

First problem

$$f_1^* = \min_{x \in \mathcal{F}} f_1(x)$$

ℓth problem

$$f_\ell^* = \min f_\ell(x)$$

subject to

$$\begin{aligned} x &\in \mathcal{F} \\ f_i(x) &= f_i^*, \; i = 1, \ldots, \ell - 1. \end{aligned}$$

Lexicographic optimization

Strong assumption

- Objectives are strictly ordered by importance.
- Any improvement in a higher-priority objective dominates all changes in lower-priority ones.
- Small modeling or measurement errors can have large effects.

ε-lexicographic optimization

Sorted objective and tolerances

- Assume that the objectives are sorted from the most important ($i = 1$) to the least important ($i = p$).
- For each $i = 1, \ldots, p$, $\varepsilon_i \geqslant 0$ is a tolerance on the objective f_i.

First problem

$$f_1^* = \min_{x \in \mathcal{F}} f_1(x)$$

ℓth problem

$$f_\ell^* = \min f_\ell(x)$$

subject to

$$\begin{aligned} x &\in \mathcal{F} \\ f_i(x) &\leqslant f_i^* + \varepsilon_i, \; i = 1, \ldots, \ell - 1. \end{aligned}$$

Outline

ε-constraints formulation

Reference objective and upper bounds

- Select a reference objective $\ell \in \{1, \ldots, p\}$.
- Impose an upper bound ε_i on each other objective.

Constrained optimization

$$\min_{x \in \mathcal{F}} f_\ell(x)$$

subject to

$$f_i(x) \leqslant \varepsilon_i, \; i \neq \ell.$$

If a solution exists, it is weakly Pareto optimal

Imposed bounds prevent improving all objectives simultaneously. However, improvements in one objective may occur without strict deterioration in others.

Outline

Local search

Main difference with single objective

Maintain a set $\mathcal{P}$ of potential Pareto optimal solutions

$$\forall x, y \in \mathcal{P}, F(x) \not\prec F(y) \text{ and } F(y) \not\prec F(x).$$

Initialization

Start with a first set $\mathcal{P}$ of candidate solutions.

Main iteration

- Select randomly x from $\mathcal{P}$ and consider x^+ a neighbor of x.
- Define
$$\mathcal{D}(x^+) = \{y \in \mathcal{P} \text{ such that } F(x^+) \prec F(y)\}.$$
- Define
$$\mathcal{S}(x^+) = \{y \in \mathcal{P} \text{ such that } F(y) \prec F(x^+)\}.$$

Local search

Main iteration

- If $\mathcal{S}(x^+) = \emptyset$

$$\mathcal{P}^+ = \mathcal{P} \cup \{x^+\} \setminus \mathcal{D}(x^+).$$

Property of $\mathcal{P}^+$

$$\forall x, y \in \mathcal{P}^+, F(x) \not\leq F(y) \text{ and } F(y) \not\leq F(x).$$

Proof

- For x, y different from x^+, already valid in $\mathcal{P}$.
- Consider x^+, $y \in \mathcal{P}^+$:
  - $y \in \mathcal{P}^+ \Rightarrow y \notin \mathcal{D}(x^+) \Rightarrow F(x^+) \not\leq F(y)$.
  - $x^+ \in \mathcal{P}^+ \Rightarrow \mathcal{S}(x^+) = \emptyset \Rightarrow y \notin \mathcal{S}(x^+) \Rightarrow F(y) \not\leq F(x^+)$.

Stopping criteria in multi-objective heuristics

Common stopping rules

- Maximum number of iterations or evaluations.
- No change in the Pareto set for a given number of iterations.
- Computational budget exhausted.

Practical note

There is no guarantee of completeness — heuristics aim at <u>approximating</u> the Pareto frontier.

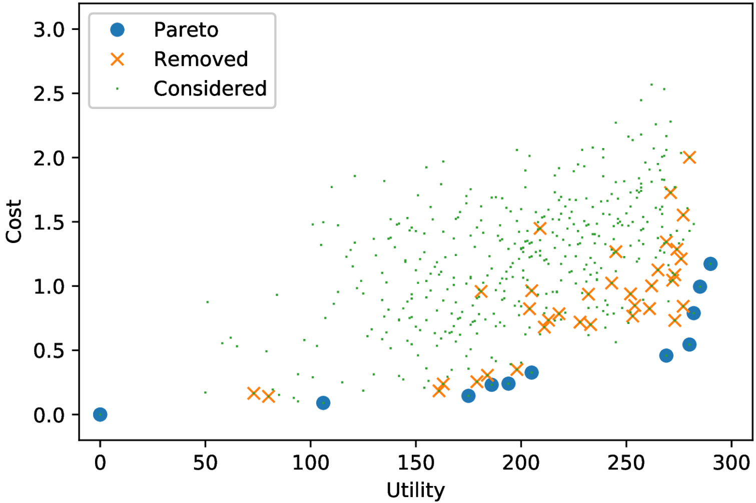
Example: priced knapsack

Utility	Weight	Cost
80	84	0.50328447
31	27	0.41431774
48	47	0.07765353
17	22	0.75842330
27	21	0.14050556
84	96	0.72089439
34	42	0.11669739
39	46	0.56723896
46	54	0.02430532
58	53	0.01255171
23	32	0.03059062
67	78	0.17285314

Objectives

- Maximize total utility,
- Minimize total cost,

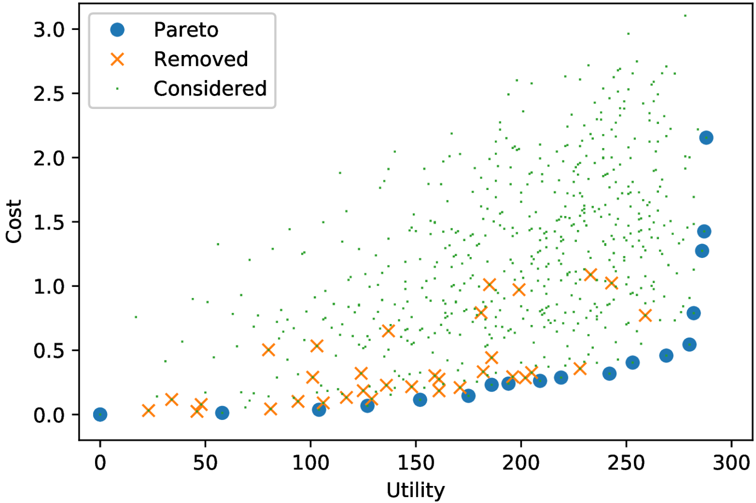
Example: local search with neighborhood $k = 4$



Variable Neighborhood Search

- Neighborhood of size 1 Pareto solutions: 1
- Neighborhood of size 2 Pareto solutions: 16
- Neighborhood of size 3 Pareto solutions: 16
- Neighborhood of size 4 Pareto solutions: 16
- Neighborhood of size 5 Pareto solutions: 16
- Neighborhood of size 6 Pareto solutions: 16
- Neighborhood of size 7 Pareto solutions: 18
- Neighborhood of size 8 Pareto solutions: 19
- Neighborhood of size 9 Pareto solutions: 19
- Neighborhood of size 10 Pareto solutions: 19
- Neighborhood of size 11 Pareto solutions: 19
- Neighborhood of size 12 Pareto solutions: 19
- Pareto solutions: 19

Variable Neighborhood Search



Conclusion

Core ideas

- Multi-objective optimization reveals trade-offs.
- Optimality is defined as a set (Pareto frontier).
- Preferences are external to optimization.

Methods

- Scalarization and constraints embed preferences.
- Heuristics approximate Pareto sets efficiently.
- Problem structure drives algorithm design.

Takeaway

Optimization supports decisions — it does not replace them.

What we did not cover

Evolutionary multi-objective algorithms

- Population-based heuristics (e.g. NSGA-II, SPEA2, MOEA/D).
- Aim to approximate the Pareto frontier in a single run.
- Rely on the same core ideas:
  - Pareto dominance,
  - trade-off between convergence and diversity,
  - heuristic exploration of the solution space.

Interactive multi-objective optimization

- Explicit interaction with a decision-maker during optimization.
- Preferences are progressively refined.
- Useful when preferences are vague, evolving, or difficult to formalize.

Takeaway

These methods differ mainly in implementation, not in fundamental concepts.

Why they were not included

Focus of this lecture

- Emphasis on concepts, not on catalogues of algorithms.
- Central questions:
  - What does optimality mean with several objectives?
  - Why does optimization return a set of solutions?
  - Where do preferences enter the process?

What you should retain

- Dominance and Pareto optimality are universal notions.
- Scalarization and constraints encode preferences.
- Heuristics approximate Pareto sets efficiently.

Key message

If you understand Pareto optimality and preference handling, you understand the foundations of all multi-objective optimization methods.