Polytechnique Montréal
Department of Mathematics and Industrial Engineering

Introduction to Multiobjective Optimization

Ludovic Salomon

Prerequites

- Basic notions in numerical analysis: gradient, hessian, vector space.
- Basic notions in optimization: solution, objective function, constraints, or convexity.

Objectives

At the end of this tutorial, you will be able to:

- Describe what a solution in multiobjective optimization is; and its main properties.
- List some general multiobjective resolution methods.

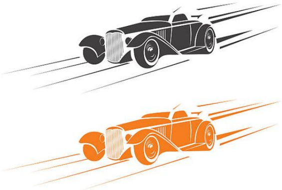
How to choose a car ? (drawn from [Ehrgott, 2005])

Your dilemna

How to choose a car ? (drawn from [Ehrgott, 2005])

Your dilemna

Your criteria



How to choose a car ? (drawn from [Ehrgott, 2005])

A market study (drawn from [Ehrgott, 2005])

Brand	VW	Opel	Ford	Toyota
Price ($)	16200	14900	**14000**	15200
Consumption (l/100 km)	7.2	**7.0**	7.5	8.2
Power (kW)	66.0	62.0	55.0	**71.0**

Conclusion

No choice is optimal : life is about making compromises ...

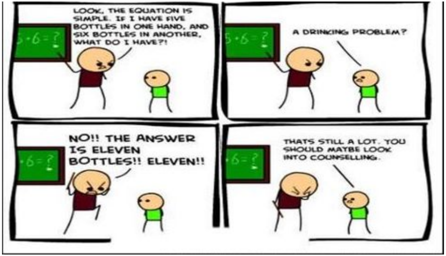
Applications

Various applications of multiobjective optimization

- Machine learning [Alexandropoulos et al., 2019, Jin, 2006]
- Chemical engineering [Sharma and Rangaiah, 2013]
- Bioinformatics and computational biology [Handl et al., 2007]
- Discrete optimization [Ehrgott, 2005]
- Economics [Tapia and Coello, 2007]
- And so on ...

Last remarks before to start

Pros



Last remarks before to start

Cons

Hard, hard, hard... More difficult to solve than single-objective optimization problems !

The general problem

The problem

$$\begin{array}{ll} \text{minimize} & f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^\top \\ x \in \Omega & \end{array}$$

where:

- $f : \Omega \to \mathbb{R}^m$ is the objective function, composed of $m \geq 2$ **objective functions** $f_1, f_2, \ldots, f_m$ for $i = 1, 2, \ldots, m$.
- $\Omega \subseteq \mathbb{R}^n$ is the **feasible decision space**.
- $f(\Omega)$ is designed as the **feasible objective space**.
- $\mathbb{R}^n$ is designed as the **decision space**, $\mathbb{R}^m$ as the **objective space**.

Order relations : definitions

Pareto dominance

Given two decision vectors x^1 and x^2,

- x^1 **weakly dominates** x^2 (denoted as $x^1 \preceq x^2$) if and only if $f_i(x^1) \leq f_i(x^2)$ for all $i = 1, 2, \ldots, m$.
  ex : when $f(x^1) = [2, 1]^\top$ and $f(x^2) = [2, 2]^\top$, $x^1 \preceq x^2$.
- x^1 **dominates** x^2 (denoted as $x^1 \prec x^2$) if and only if $f_i(x^1) \leq f_i(x^2)$ for all $i = 1, 2, \ldots, m$ and there exists index $j \in \{1, 2, \ldots, m\}$ such that $f_j(x^1) < f_j(x^2)$.
  ex : for $f(x^1) = [1, 0]^\top$ and $f(x^2) = [1, 1]^\top$, $x^1 \prec x^2$.
- x^1 **strictly dominates** x^2 (denoted as $x^1 \prec\prec x^2$) if and only if $f_i(x^1) < f_i(x^2)$ for all $i = 1, 2, \ldots, m$.
  ex : when $f(x^1) = [1, 0]^\top$ and $f(x^2) = [5, 6]^\top$, $x^1 \prec\prec x^2$.
- x^1 and x^2 are **incomparable** (denoted as $x^1 \parallel x^2$) if and only if x^1 does not weakly dominate x^2 nor x^2 does not weakly dominate x^2.
  ex : when $f(x^1) = [-1, 3]^\top$ and $f(x^2) = [7, -2]^\top$, $x^1 \parallel x^2$.

Dominance, dominated and indifferent zones

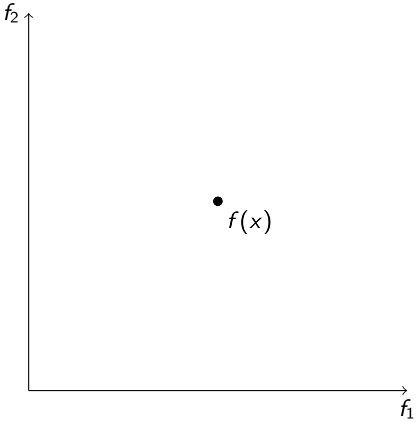


Figure: Principal search zones for a biobjective minimization problem in the objective space.

Dominance, dominated and indifferent zones

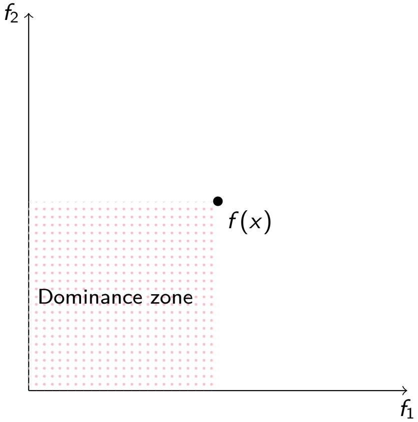


Figure: Principal search zones for a biobjective minimization problem in the objective space.

Dominance, dominated and indifferent zones

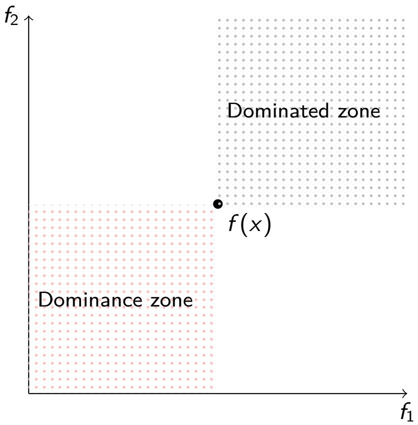


Figure: Principal search zones for a biobjective minimization problem in the objective space.

Dominance, dominated and indifferent zones

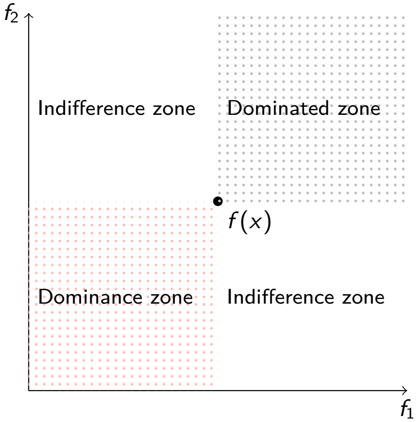


Figure: Principal search zones for a biobjective minimization problem in the objective space.

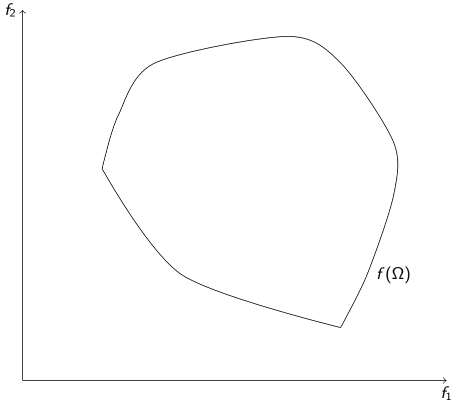
Pareto front and Pareto set

Definition

The decision vector $x^* \in \Omega$ is said to be **Pareto-optimal** if there does not exist any other decision vector $x \in \Omega$ such that $x \prec x^*$.
The set of all Pareto optimal solutions is called the **Pareto set** denoted as $\mathcal{X}_P$ and its image by the objective function is called the **Pareto front** denoted as $\mathcal{Y}_P$.

Pareto front and Pareto set

An illustration,



Pareto front and Pareto set

An illustration,

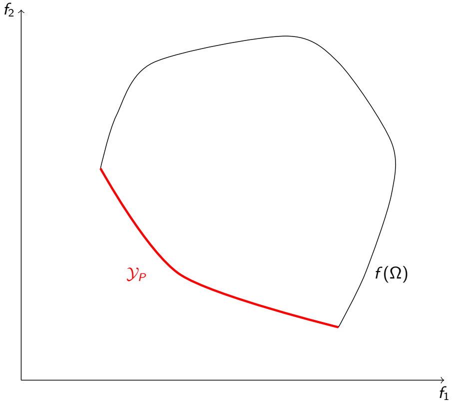
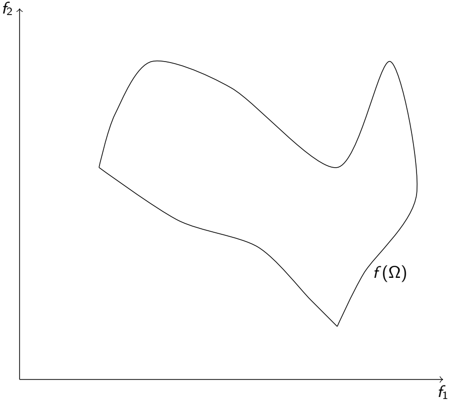


Figure: Objective space and convex Pareto front for a biobjective minimization problem.

Pareto front and Pareto set

Another illustration,



Pareto front and Pareto set

Another illustration,

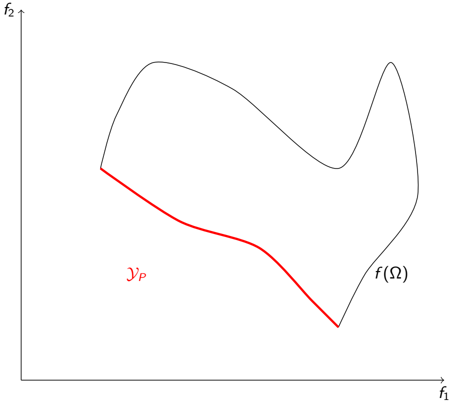
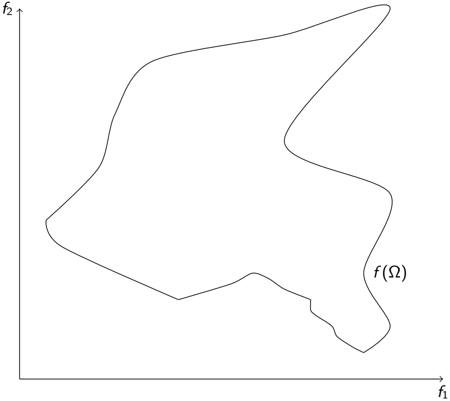


Figure: Objective space and non convex Pareto front for a biobjective minimization problem.

Pareto front and Pareto set

And a last one !



Pareto front and Pareto set

And a last one !

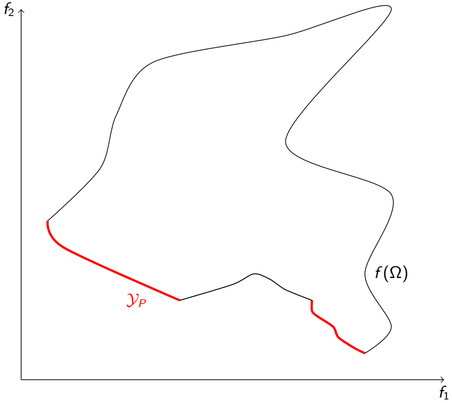


Figure: Objective space and piecewise-continuous Pareto front for a biobjective minimization problem

Bounds on the Pareto front

Definition

The **ideal objective vector** y^I is defined as

$$y^I = \left[\min_{x\in\Omega} f_1(x), \min_{x\in\Omega} f_2(x), \ldots, \min_{x\in\Omega} f_m(x)\right]^\top$$

Definition

The **nadir objective vector** y^N is defined as

$$y^N = \left[\max_{x\in\mathcal{X}_P} f_1(x), \max_{x\in\mathcal{X}_P} f_2(x), \ldots, \max_{x\in\mathcal{X}_\mathcal{P}} f_m(x)\right]^\top$$

Definition

The **utopian objective vector** y^U is defined as

$$y^U = \left[y_1^I - \epsilon_1, y_2^I - \epsilon_2, \ldots, y_m^I - \epsilon_m\right]^\top$$

where $\epsilon_i > 0$ for $i = 1, 2, \ldots, m$.

Bounds on the Pareto front

An illustration

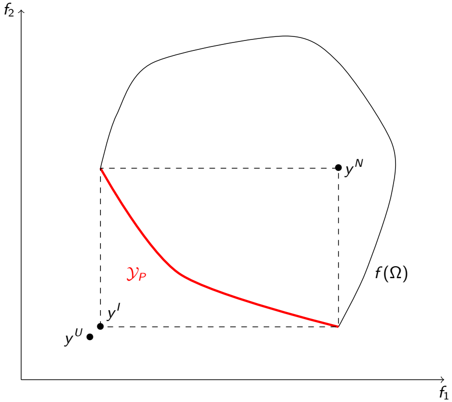


Figure: Objective space, ideal, utopian and nadir objective vector for a biobjective minimization problem.

Limits

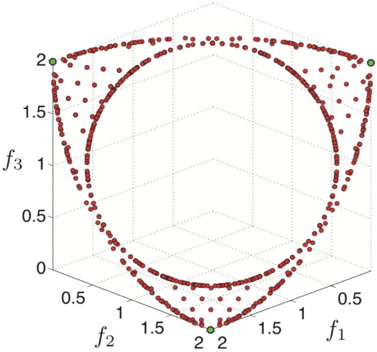


Figure: An exemple of a Pareto front approximation [Burachik et al., 2017].

Scalarization methods: the main idea

- Rely on single-objective optimization methods.
- Convert the multiobjective optimization problem (MOP) into a succession of parameterized single-objective subproblems (SOP_j).

Goal

Solve $(SOP_j) \Rightarrow$ One non dominated point.

Scalarization methods: the main idea

- Rely on single-objective optimization methods.
- Convert the multiobjective optimization problem (MOP) into a succession of parameterized single-objective subproblems (SOP_j).

Goal

$$\text{Solve } (SOP_j) \Rightarrow \text{One non dominated point.}$$

The Weighted-sum scalarization method [Miettinen, 1999]

Principle

Given a set of positive weights w_i for $i = 1, 2, \ldots, m$ such that

$$\sum_{i=1}^{m} w_i = 1,$$

solve the subproblem

$$\underset{x \in \Omega}{\text{minimize}} \quad \sum_{i=1}^{m} w_i f_i(x)$$

Pros

- Intuitive to understand and easy to interpret.
- The global solution of this problem is weakly Pareto optimal.

The Weighted-sum scalarization method [Miettinen, 1999]

Principle

Given a set of positive weights w_i for $i = 1, 2, \ldots, m$ such that

$$\sum_{i=1}^{m} w_i = 1,$$

solve the subproblem

$$\underset{x \in \Omega}{\text{minimize}} \quad \sum_{i=1}^{m} w_i f_i(x)$$

Pros

- Intuitive to understand and easy to interpret.
- The global solution of this problem is weakly Pareto optimal.

The Weighted-sum scalarization method [Miettinen, 1999]

Cons

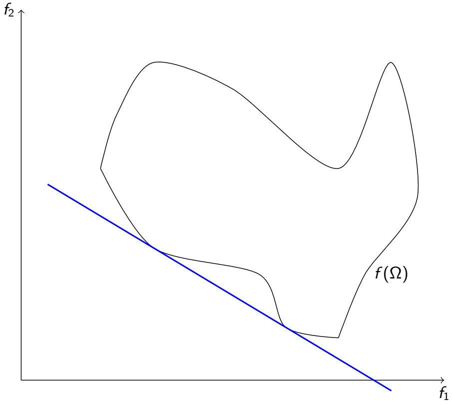


Figure: The weighted sum approach may generate only a subset of the Pareto front.

The ε-constraint method [Chankong and Haimes, 1983]

Principle

Solve the following subproblem

$$\begin{array}{ll} \text{minimize} & f_l(x) \\ x \in \Omega(\varepsilon) & \end{array}$$

where

$$\Omega(\varepsilon) = \{x \in \Omega : f_j(x) \leq \varepsilon_j,\ j = 1, 2, \ldots, m,\ j \neq l\}$$

The ε-constraint method [Chankong and Haimes, 1983]

Functioning

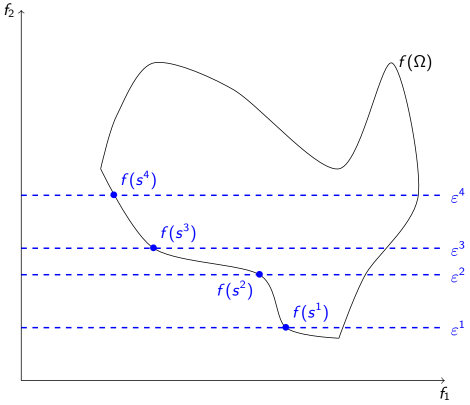


Figure: Generation of non dominated points using the ε constraint method.

The ε-constraint method [Chankong and Haimes, 1983]

Pros

- Easy to understand.
- Easy to interpret.

Cons

- Numerical issues.
- Difficulties to solve the single-objective optimization subproblem.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Principle

Solve the following subproblem

$$\begin{array}{ll} \text{maximize} & t \\ \text{s.t} & f(x) = \Phi w + tn \\ & x \in \Omega, t \in \mathbb{R} \end{array}$$

where:

- w is a vector of weights: $w_i \geq 0$ and $\sum_{i=1}^{m} w_i = 1$.
- n is an unit vector orthogonal to the CHIM simplex (Convex Hull of Individual Minima) pointing to the origin.
- Φ the $m \times m$ matrix whose columns are $f(x^{j,\star}) - y^I$ with

$$x^{j,\star} \in \arg\min_{x \in \Omega} f_j(x).$$

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

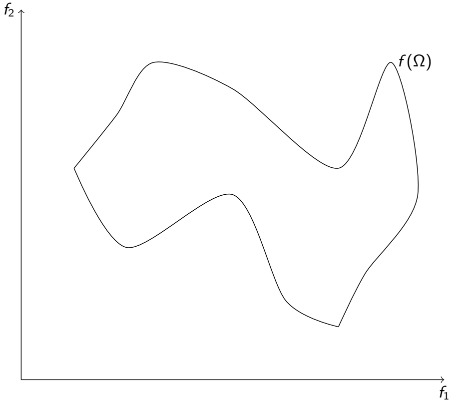


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

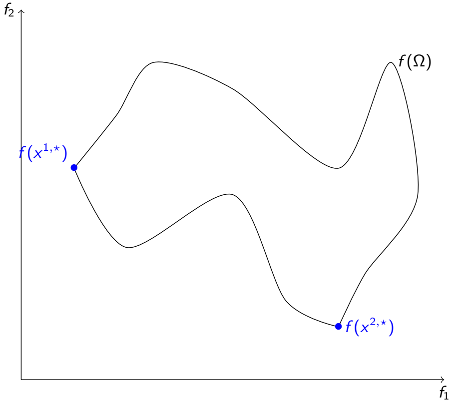


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

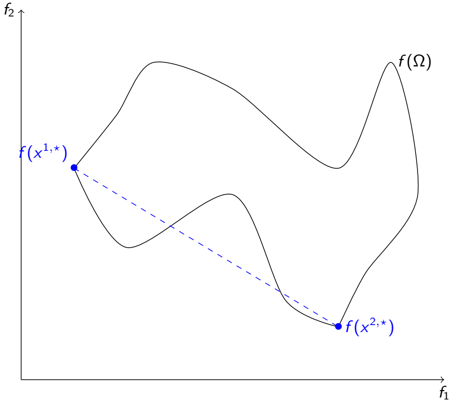


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

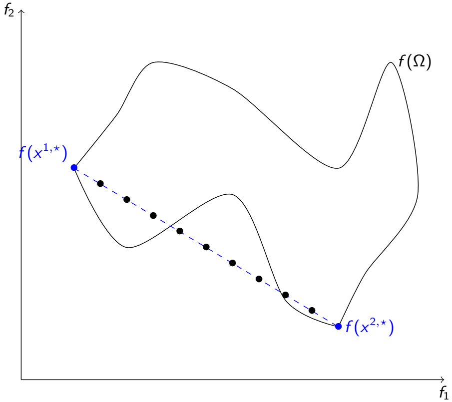


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

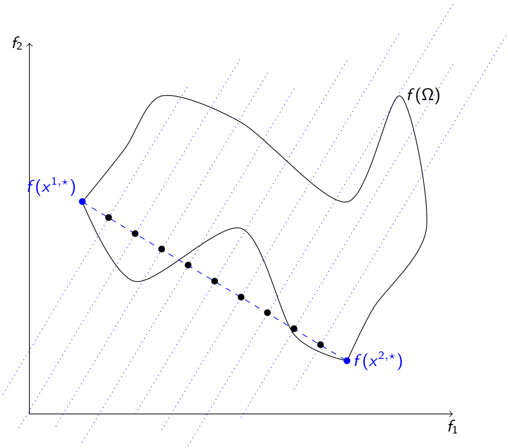


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

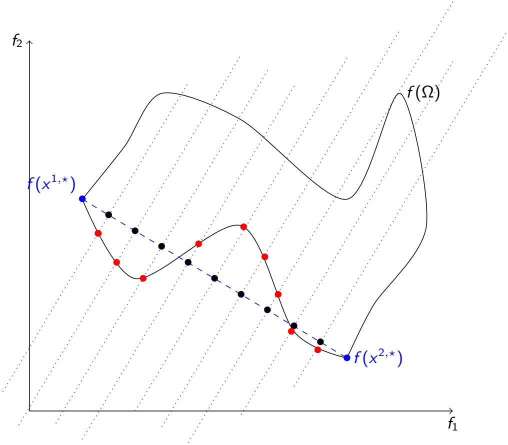


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

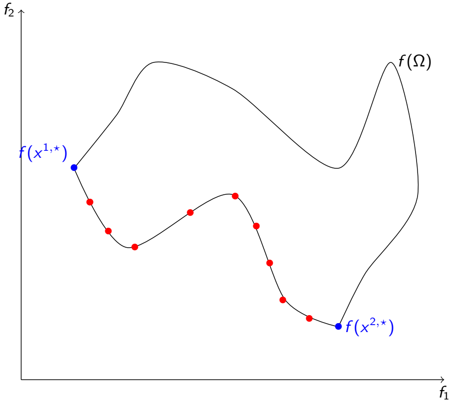


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Illustration

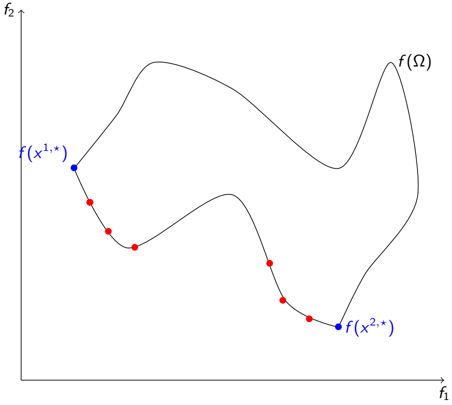


Figure: Generation of non dominated points using the NBI method.

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

A limitation of NBI

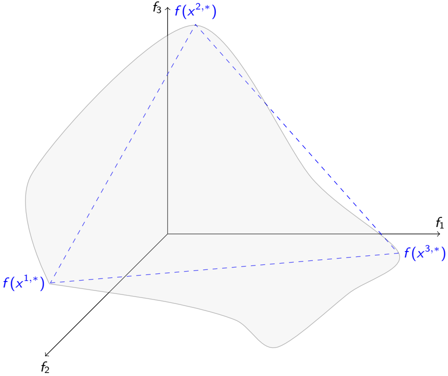


Figure: The NBI method can miss some parts of the Pareto front (inspired by [Das and Dennis, 1998]).

The Normal Boundary Intersection (NBI) method [Das and Dennis, 1998]

Pros

- Generate a good distribution of Pareto objective vectors
- Adaptive for many types of problems.

Cons

- The equality constraints can complexify the resolution of the subproblem.
- The method can generate dominated points.
- Can miss some part of the Pareto front for more than 2 objectives [Das and Dennis, 1998].

Other Variant

- Normal Constraint Method [Messac and Mattson, 2004]

Weighted Tchebycheff methods [Miettinen, 1999]

Weighted Tchebycheff method

Given a set of weights w_i for $i = 1, 2, \ldots, m$ with $\sum_{i=1}^{m} w_i = 1$, solve the subproblem

$$\begin{array}{ll} \text{minimize} & \max\limits_{1 \le i \le m} w_i |f_i(x) - r_i| \\ x \in \Omega & \end{array}$$

where $r \in \mathbb{R}^m$ is a reference objective vector. Generally, $r \in \{y^I, y^U\}$.

Weighted Tchebycheff methods [Miettinen, 1999]

Weighted Tchebycheff augmented methods

Given a set of weights w_i for $i = 1, 2, \ldots, m$ with $\sum_{i=1}^{m} w_i = 1$, solve the subproblem

$$\begin{array}{ll} \text{minimize} & \left[\max\limits_{1\leq i\leq m} w_i(f_i(x) - r_i)\right] + \rho \sum_{i=1}^{m} f_i(x) - r_i \\ x \in \Omega & \end{array}$$

where

- $r \in \mathbb{R}^m$ is a reference objective vector. Generally, $r \in \{y^I, y^U\}$.
- $\rho > 0$ is an external parameter.

Another variant is

$$\begin{array}{ll} \text{minimize} & \max\limits_{1\leq i\leq m} w_i \left[f_i(x) - r_i + \rho \sum\limits_{i=1}^{m} f_i(x) - r_i\right] \\ x \in \Omega & \end{array}$$

Weighted Tchebycheff methods [Miettinen, 1999]

Pros

- Can capture non convex parts of the Pareto front.

Cons

- Choice of the weights
- Require a reformulation.

The goal programming scalarization method

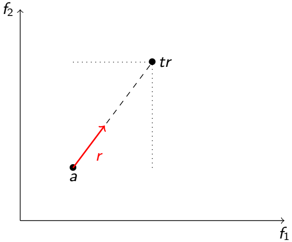
Pascoletti and Serafini scalarization [Pascoletti and Serafini, 1984]

Solve the following subproblem

$$\begin{array}{ll} \text{minimize} & t \\ \text{s.t.} & f(x) \leq a + tr \\ & (x,t) \in \Omega \times \mathbb{R} \end{array}$$

with $r \in \mathbb{R}^m$ and $a \in \mathbb{R}^m$.

Interpretation



The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

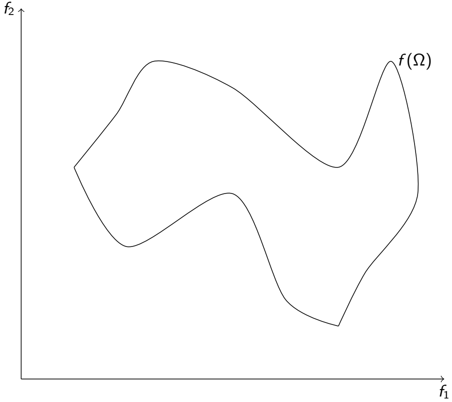


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

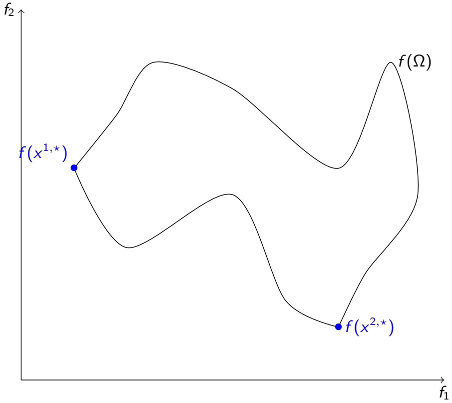


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

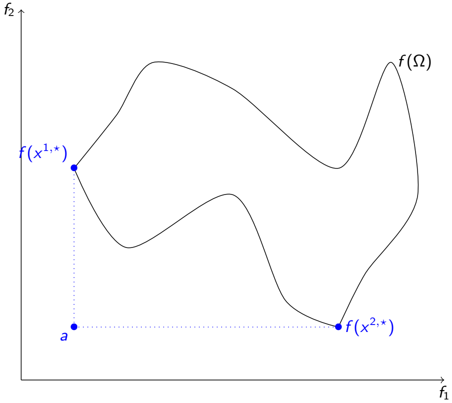


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

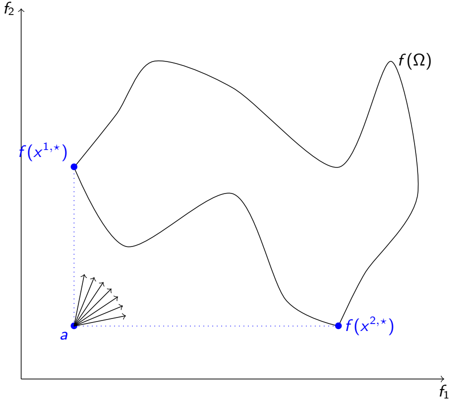


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

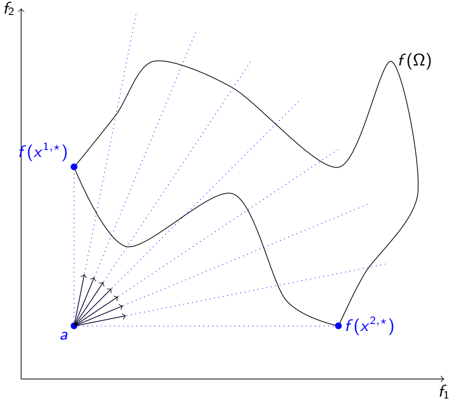


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

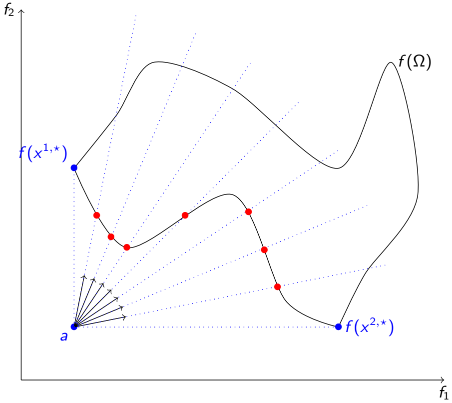


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

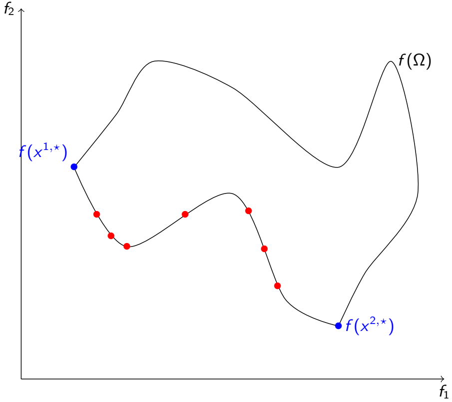


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

The goal programming scalarization method

Illustration (inspired by [Ghosh and Chakraborty, 2015, Khorram et al., 2014])

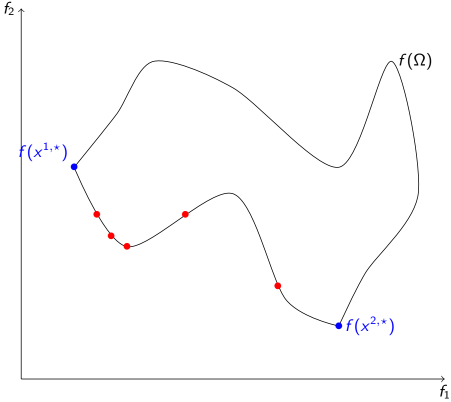


Figure: Generation of non dominated points using the Pascoletti and Serafini approach.

To go further

- Some general references on scalarization methods [Miettinen, 1999, Wiecek et al., 2016].
- How to deal with $m \geq 3$ objectives ? Some works [Mueller-Gritschneder et al., 2009, Burachik et al., 2017] explore this path.
- Recently, methods using a combination of scalarization approaches and branch and bound techniques have been proposed [Eichfelder et al., 2021, Niebling and Eichfelder, 2019].

A reminder

The problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a **scalar-valued** function and continuously differentiable.

A reminder

The problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a **scalar-valued** function and continuously differentiable.

A simple minimization optimization algorithm

- **Initialization** : choose a starting point $x^0 \in \mathbb{R}^n$.
- **Main loop** : for $k = 0, 1, \dots$
  1. If x^k satisfies a stopping condition, then stop.
  2. Otherwise, choose a descent direction d^k.
  3. Choose a step length $\alpha^k > 0$.
  4. Set $x^{k+1} := x^k + \alpha^k d^k$.

A reminder

The problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a **scalar-valued** function and continuously differentiable.

Some interesting descent directions

- The **steepest descent direction**

$$d^k = -\nabla f(x^k).$$

- The **Newton direction**

$$d^k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

- The **Quasi-Nexton direction**

$$d^k = -(B^k)^{-1} \nabla f(x^k)$$

  where $B^k \in \mathbb{R}^n$ positive symmetric.

A reminder

Question

Can we define descent directions for multiobjective optimization problems ?

Steepest descent direction for unconstrained multiobjective optimization

The problem

$$\min_{x\in\mathbb{R}^n} f(x)$$

where $f:\mathbb{R}^n \to \mathbb{R}^m$ is **continuously differentiable**.

Steepest descent direction for unconstrained multiobjective optimization

The problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$ is **continuously differentiable**.

Theorem

If $x^* \in \mathbb{R}^n$ is Pareto optimal, then there does not exist any direction $d \in \mathbb{R}^n$ such that for all indexes $i = 1, 2, \ldots, m$,

$$\nabla f_i(x)^T d < 0.$$

Idea

Study

$$\max_{i=1,2,\ldots,m} \nabla f_i(x)^T d.$$

Steepest descent direction for unconstrained multiobjective optimization

The problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$ is **continuously differentiable**.

Steepest gradient descent for unconstrained multiobjective optimization [Fliege and Svaiter, 2000]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\dots,m} \nabla f_i(x)^T d + \frac{1}{2}\|d\|^2$$

Steepest descent direction for unconstrained multiobjective optimization

The problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$ is **continuously differentiable**.

Steepest gradient descent for unconstrained multiobjective optimization [Fliege and Svaiter, 2000]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\ldots,m} \nabla f_i(x)^T d + \frac{1}{2}\|d\|^2$$

Remarks

- The problem can be reformulated as a smooth one:

$$\begin{array}{ll} \text{minimize} & t + \frac{1}{2}\|d\|^2 \\ \text{s.t.} & \nabla f_i(x)^\top d \leq t, i = 1, 2, \ldots, m \\ & t \in \mathbb{R},\ x \in \mathbb{R}^n \end{array}$$

- If $m = 1$, one gets $d^k = -\nabla f(x^k)$.

Other descent directions for unconstrained multiobjective optimization

Assumption

$f : \mathbb{R}^n \to \mathbb{R}^m$ is $\mathcal{C}^2$, i.e. **twice differentiable** on $\mathbb{R}^n$.

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\ldots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\ldots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Quasi-Newton direction for unconstrained multiobjective optimization [Morovati et al., 2017, Qu et al., 2011]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\ldots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top B^{k,i} d$$

where $B^{k,i} \in \mathbb{R}^{n \times n}$ for $i = 1, 2, \ldots, m$ is symmetric.

Other descent directions for unconstrained multiobjective optimization

Assumption

$f : \mathbb{R}^n \to \mathbb{R}^m$ is $\mathcal{C}^2$, i.e. **twice differentiable** on $\mathbb{R}^n$.

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\dots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\dots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Quasi-Newton direction for unconstrained multiobjective optimization [Morovati et al., 2017, Qu et al., 2011]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\dots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top B^{k,i} d$$

where $B^{k,i} \in \mathbb{R}^{n \times n}$ for $i = 1, 2, \dots, m$ is symmetric.

Other descent directions for unconstrained multiobjective optimization

Assumption

$f : \mathbb{R}^n \to \mathbb{R}^m$ is $\mathcal{C}^2$, i.e. **twice differentiable** on $\mathbb{R}^n$.

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg \min_{d \in \mathbb{R}^n} \max_{i=1,2,\dots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg \min_{d \in \mathbb{R}^n} \max_{i=1,2,\dots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Quasi-Newton direction for unconstrained multiobjective optimization [Morovati et al., 2017, Qu et al., 2011]

$$d(x) \in \arg \min_{d \in \mathbb{R}^n} \max_{i=1,2,\dots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top B^{k,i} d$$

where $B^{k,i} \in \mathbb{R}^{n \times n}$ for $i = 1, 2, \dots, m$ is symmetric.

Other descent directions for unconstrained multiobjective optimization

Assumption

$f : \mathbb{R}^n \to \mathbb{R}^m$ is $\mathcal{C}^2$, i.e. **twice differentiable** on $\mathbb{R}^n$.

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\ldots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Newton direction for unconstrained multiobjective optimization [Fliege et al., 2009]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\ldots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top \nabla^2 f_i(x) d.$$

Quasi-Newton direction for unconstrained multiobjective optimization [Morovati et al., 2017, Qu et al., 2011]

$$d(x) \in \arg\min_{d \in \mathbb{R}^n} \max_{i=1,2,\ldots,m} \nabla f_i(x)^\top d + \frac{1}{2} d^\top B^{k,i} d$$

where $B^{k,i} \in \mathbb{R}^{n \times n}$ for $i = 1, 2, \ldots, m$ is symmetric.

Advantages and inconveniences

Pros

- Do not require external parameters.
- Intuitive to understand.

Cons

- The subproblem can be difficult to solve.
- Generate only one non dominated point given a starting point.

Heuristics

Principles

- Start from an initial population of points.
- Interaction between them defined by some parameters.
- Mutation and selection along iterations, supposedly toward some Pareto points.
- For more information, see [Deb and Miettinen, 2008].

Classical methods

- Evolutionary algorithms: NSGA-II [Deb et al., 2000], MOEA-D [Zhang and Li, 2007], ...
- Particule-swarm optimization [Poli et al., 2007]: ant, wolf, butterfly, ...

Heuristics

Some software

- jMetal [Durillo and Nebro, 2011, Nebro et al., 2015]: https://github.com/jMetal/jMetal.
- Pagmo2 [Biscani and Izzo, 2020]: https://github.com/esa/pagmo2.
- Pymoo [Blank and Deb, 2020]: https://pymoo.org/.

Heuristics

Pros

- Really versatile (can deal with integer variables, ...).
- Tunable.

Cons

- Tunable.
- Not really efficient.
- Are really bad to deal with constraints.
- Difficult to scale with a huge number of variables.
- The population size parameter limits the number of potential solutions.

Exact black-box/derivative-free optimization methods

Algorithms

- Direct Multi Search (DMS) [Custódio et al., 2011].
- Multiobjective Implicit Filtering Optimization (MOIF) [Cocchi et al., 2018].
- Derivative-Free Multiobjective Optimization (DFMO) [Liuzzi et al., 2016].

Advantages

- Very similar to heuristics
  1. Start from an initial population of points.
  2. Update the population towards iterations.
- Do not make any restrictions on the size of the population.
- Deterministic convergence analysis.
- Efficient.

Conclusion

At the end of this tutorial, you are now able to:

- Describe what a solution in multiobjective optimization is; the principal properties of a Pareto front.
- List some general multiobjective resolution methods.

Conclusion

What I did not mention

- The construction of efficient data structures for Pareto fronts [Bentley et al., 1993, Chen et al., 2012, De et al., 2017, Jaszkiewicz and Lust, 2018]:
  see `https://alandefreitas.github.io/pareto/` for an implementation.
- The evaluation of the performance of multiobjective optimization algorithms [Audet et al., 2021, Li and Yao, 2019].
- Discrete multiobjective optimization
  ([Ehrgott, 2005, Holzmann and Smith, 2018, Kirlik and Sayın, 2014] for example).

Conclusion

Multiobjective optimization algorithms

- are a tool to **better model** engineering problems.
- are **more difficult to solve** than single-objective optimization problems.
- traditionally are dealt using single-objective optimization methods and scalarization techniques.

But

- **difficult** to implement;
- there are a **limited number** of bullet-proof available algorithms;
- hence the predominance of heuristics.
- To follow...

Thank you for your attention !
Do you have any questions ?

Alexandropoulos, S.-A. N., Aridas, C. K., Kotsiantis, S. B., and Vrahatis, M. N. (2019).
Multi-Objective Evolutionary Optimization Algorithms for Machine Learning: A Recent Survey, pages 35–55.
Springer International Publishing, Cham.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., and Salomon, L. (2021).
Performance indicators in multiobjective optimization.
European Journal of Operational Research, 292(2):397–422.

Bentley, J. L., Clarkson, K. L., and Levine, D. B. (1993).
Fast linear expected-time algorithms for computing maxima and convex hulls.
Algorithmica, 9(2):168–183.

Biscani, F. and Izzo, D. (2020).
A parallel global multiobjective framework for optimization: pagmo.
Journal of Open Source Software, 5(53):2338.

Blank, J. and Deb, K. (2020).
Pymoo: Multi-objective optimization in python.
IEEE Access, 8:89497–89509.

Burachik, R. S., Kaya, C. Y., and Rizvi, M. M. (2017).
A New Scalarization Technique and New Algorithms to Generate Pareto Fronts.
SIAM Journal on Optimization, 27(2):1010–1034.

Chankong, V. and Haimes, Y. (1983).
Multiobjective Decision Making: Theory and Methodology.

Chen, W.-M., Hwang, H.-K., and Tsai, T.-H. (2012).
Maxima-finding algorithms for multidimensional samples: A two-phase approach.
Computational Geometry, 45(1):33–53.

Cocchi, G., Liuzzi, G., Papini, A., and Sciandrone, M. (2018).
An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints.
Computational Optimization and Applications, 69(2):267–296.

Custódio, A. L., Madeira, J. F. A., Vaz, A. I. F., and Vicente, L. N. (2011).
Direct multisearch for multiobjective optimization.
SIAM Journal on Optimization, 21(3):1109–1140.

Das, I. and Dennis, J. E. (1998).
Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems.
SIAM Journal on Optimization, 8(3):631–657.

De, S. S., Dehuri, S., and Cho, S.-B. (2017).
Divide-and-conquer approach for revealing the non-dominated solutions in multi-objective optimization problem.
In *2017 International Conference on Information Technology (ICIT)*, pages 143–151. IEEE.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000).
A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.
In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature PPSN VI*, pages 849–858, Berlin, Heidelberg. Springer Berlin Heidelberg.

Deb, K. and Miettinen, K. (2008).
Multiobjective optimization: interactive and evolutionary approaches, volume 5252.
Springer Science & Business Media.

Durillo, J. J. and Nebro, A. J. (2011).
jmetal: A java framework for multi-objective optimization.
Advances in Engineering Software, 42(10):760–771.

Ehrgott, M. (2005).
Multicriteria Optimization, volume 491.
Springer-Verlag.

Eichfelder, G., Kirst, P., Meng, L., and Stein, O. (2021).
A general branch-and-bound framework for continuous global multiobjective optimization.
Journal of Global Optimization.

Fliege, J., Drummond, L. M. G., and Svaiter, B. F. (2009).
Newton's method for multiobjective optimization.
SIAM Journal on Optimization, 20(2):602–626.

Fliege, J. and Svaiter, B. F. (2000).
Steepest descent methods for multicriteria optimization.
Mathematical Methods of Operations Research, 51(3):479–494.

Ghosh, D. and Chakraborty, D. (2015).
A direction based classical method to obtain complete pareto set of multi-criteria optimization problems.
Opsearch, 52(2):340–366.

Handl, J., Kell, D. B., and Knowles, J. (2007).
Multiobjective optimization in bioinformatics and computational biology.
IEEE/ACM Trans. Comput. Biol. Bioinformatics, 4(2):279–292.

Holzmann, T. and Smith, J. (2018).
Solving discrete multi-objective optimization problems using modified augmented weighted tchebychev scalarizations.
European Journal of Operational Research, 271(2):436–449.

Jaszkiewicz, A. and Lust, T. (2018).
ND-tree-based update: A fast algorithm for the dynamic nondominance problem.
IEEE Transactions on Evolutionary Computation, 22(5):778–791.

Jin, Y. (2006).
Multi-Objective Machine Learning, volume 16.
Springer Science & Business Media.

Khorram, E., Khaledian, K., and Khaledyan, M. (2014).
A numerical method for constructing the Pareto front of multi-objective optimization problems.
Journal of Computational and Applied Mathematics, 261:158–171.

Kirlik, G. and Sayın, S. (2014).
A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems.
European Journal of Operational Research, 232(3):479–488.

Li, M. and Yao, X. (2019).
Quality evaluation of solution sets in multiobjective optimisation: A survey.
ACM Comput. Surv., 52(2):26:1–26:38.

Liuzzi, G., Lucidi, S., and Rinaldi, F. (2016).
A derivative-free approach to constrained multiobjective nonsmooth optimization.
SIAM Journal on Optimization, 26(4):2744–2774.

Messac, A. and Mattson, C. A. (2004).
Normal constraint method with guarantee of even representation of complete pareto frontier.
AIAA Journal, 42(10):2101–2111.

Miettinen, K. (1999).
Nonlinear multiobjective optimization, volume 12.
Springer Science & Business Media.

Morovati, V., Basirzadeh, H., and Pourkarimi, L. (2017).
Quasi-Newton methods for multiobjective optimization problems.
4OR, 16(3):261–294.

Mueller-Gritschneder, D., Graeb, H., and Schlichtmann, U. (2009).
A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems.
SIAM Journal on Optimization, 20(2):915–934.

Nebro, A. J., Durillo, J. J., and Vergne, M. (2015).
Redesigning the jmetal multi-objective optimization framework.
In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '15, pages 1093–1100, New York, NY, USA. Association for Computing Machinery.

Niebling, J. and Eichfelder, G. (2019).
A branch–and–bound-based algorithm for nonconvex multiobjective optimization.
SIAM Journal on Optimization, 29(1):794–821.

Pascoletti, A. and Serafini, P. (1984).
Scalarizing vector optimization problems.
Journal of Optimization Theory and Applications, 42(4):499–524.

Poli, R., Kennedy, J., and Blackwell, T. (2007).
Particle swarm optimization.
Swarm Intelligence, 1(1):33–57.

Qu, S., Goh, M., and Chan, F. T. (2011).
Quasi-newton methods for solving multiobjective optimization.
Operations Research Letters, 39(5):397–399.

Sharma, S. and Rangaiah, G. P. (2013).
Multi-Objective Optimization Applications in Chemical Engineering, chapter 3, pages 35–102.
John Wiley & Sons, Ltd.

Tapia, M. G. C. and Coello, C. A. C. (2007).
Applications of multi-objective evolutionary algorithms in economics and finance: A survey.
In *2007 IEEE Congress on Evolutionary Computation*, pages 532–539. IEEE.

Wiecek, M. M., Ehrgott, M., and Engau, A. (2016).
Continuous multiobjective programming.
In *Multiple Criteria Decision Analysis*, pages 739–815. Springer New York.

Zhang, Q. and Li, H. (2007).
Moea/d: A multiobjective evolutionary algorithm based on decomposition.
IEEE Transactions on Evolutionary Computation, 11(6):712–731.