DM872
Mathematical Optimization at Work

# Multiobjective Optimization

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

# Outline

# Multi-Objective Optimization: Basics

Multiobjective optimization problem with $p$ objective functions to be minimized over a discrete, nonempty, set $X$ of feasible solutions:

$$\min \{f_1(\boldsymbol{x}), \ldots, f_p(\boldsymbol{x}) \mid \boldsymbol{x} \in X\}$$

- $\boldsymbol{x} \in X$ a feasible solution, $\boldsymbol{y} = [f_1(\boldsymbol{x}), \ldots, f_p(\boldsymbol{x})]$ a feasible point (image)
- $Y = f(X) \subseteq \mathbb{R}^p$ set of images of all feasible solutions in objective space.
  Assume $Y$ is bounded. We can rewrite:

  $$\min \{(y_1 \ldots, y_p) \mid \boldsymbol{y} \in Y\}$$

- Given a point $\boldsymbol{y} \in \mathbb{R}^p$, $\boldsymbol{y}_{-k}$ is its orthogonal projection on the subspace $\mathbb{R}^{p-1}$ i.e.

  $$\boldsymbol{y}_{-k} = (y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_p)$$

# Multi-Objective Optimization: Basics

- Objective vector $\boldsymbol{f} = (f_1, \ldots, f_p)$
- We want to minimize $\boldsymbol{f}$ but what does it mean?
  - Scalarization (eg, Weighted sum)
  - Lexicographic
  - Pareto optimality (without previous knwoledge on component importance)

# Basic Notions

Partial orders on $\mathbb{R}^p$: let $\boldsymbol{u}$ and $\boldsymbol{v}$ be vectors in $\mathbb{R}^p$:

| | | |
|---|---|---|
| weak component-wise order | $\boldsymbol{u} \leqq \boldsymbol{v}$ | $u_i \leq v_i,\ i = 1, \ldots, p$; |
| component-wise order | $\boldsymbol{u} \preceq \boldsymbol{v}$ | $u_i \leq v_i,\ i = 1, \ldots, p$ and $\boldsymbol{u} \neq \boldsymbol{v}$; |
| strict component-wise order | $\boldsymbol{u} \prec \boldsymbol{v}$ | $u_i < v_i,\ i = 1, \ldots, p$ |

Let $\boldsymbol{x}$ and $\boldsymbol{x}'$ be two feasible solutions and $\boldsymbol{y} = f(\boldsymbol{x})$ and $\boldsymbol{y}' = f(\boldsymbol{x}')$:

| | |
|---|---|
| $\boldsymbol{y}$ weakly dominates $\boldsymbol{y}'$ | iff $\boldsymbol{y} \leqq \boldsymbol{y}'$ |
| $\boldsymbol{y}$ (Pareto) dominates $\boldsymbol{y}'$ | iff $\boldsymbol{y} \preceq \boldsymbol{y}'$ |
| $\boldsymbol{y}$ strictly dominates $\boldsymbol{y}'$ | iff $\boldsymbol{y} \prec \boldsymbol{y}'$ |

- $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ are non-weakly dominated if $f(\boldsymbol{x}) \nleqq f(\boldsymbol{x}')$ and $f(\boldsymbol{x}') \nleqq f(\boldsymbol{x})$
- $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ are non-dominated if $f(\boldsymbol{x}) \npreceq f(\boldsymbol{x}')$ and $f(\boldsymbol{x}') \npreceq f(\boldsymbol{x})$

- A solution $\boldsymbol{x}$ is called efficient (or Pareto global optimum solution) iff its image is not dominated: there is no $\boldsymbol{x}' \in X$ such that $\boldsymbol{f}(\boldsymbol{x}') \preceq \boldsymbol{f}(\boldsymbol{x})$
- A solution $\boldsymbol{x}$ is called weakly efficient iff its image is not strictly dominated.

# Basic Notions

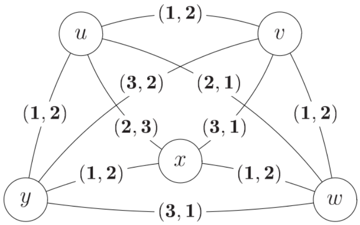
- A set of solutions $S$ is a Pareto global optimum set iff it contains only and all Pareto global optimum solutions.

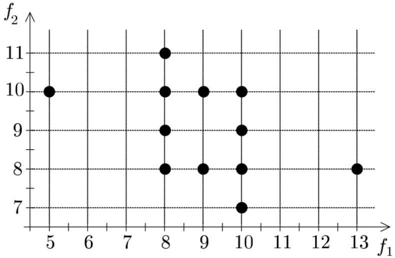
- Efficient set (or Pareto frontier) is the image of the Pareto global optimum set in the objective space.

- $X^* \subseteq X$ is strict Pareto global optimum set iff:
  - it contains only Pareto global optimum solutions
  - the corresponding set of objective function value vectors coincides with the efficient set and its elements are unique.

Let $Y_{ND} \subseteq Y$ be the set of points that are non-dominated (ie, the efficient set). We want an algorithm to generate $Y_{ND}$ and provide one of the corresponding efficient solutions for each point of this set.

# Example – Multi-objective TSP




Pareto global optimum set:

| $\pi$ | $f(\pi)$ |
|---|---|
| $\pi = [u, v, w, x, y]$ | $[5, 10]$ |
| $\pi = [u, w, v, x, y]$ | $[8, 8]$ |
| $\pi = [u, v, w, x, y]$ | $[10, 7]$ |

strict Pareto global optimum set

if all edges had weights, eg, $(3, 3)$, then all $\binom{5}{2}$ solutions would have cost $[5 \cdot 3, 5 \cdot 3]$ and would be in the Pareto global optimum set. However, both the efficient set and the strict Pareto global optimum set would have one single solution, which is any of the feasible ones.

# Basic Notions

Computational class #P: concerned with counting the number of solutions.

A counting problem belongs to #P if there is a polynomial nondeterministic algorithm such that for any instance of the problem, it computes a number of yes-answers that is equal to the number of distinct solutions of that instance.

Class #P-complete: a problem $\mathcal{P}_1$ is #P-complete if it belongs to #P and for all problems $\mathcal{P}_2$ in #P there exists a polynomial transformation from $\mathcal{P}_1$ to $\mathcal{P}_2$ such that any instance of $\mathcal{P}_1$ is mapped into an instance of $\mathcal{P}_2$ with the same number of yes-answers as the instance of $\mathcal{P}_1$.

# Pareto Optimal Solutions

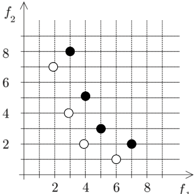
How do we find the set of Pareto Optimal solutions?

- evolutionary algorithms
- scalarization method
- enumeration (branch and bound and dynamic programming) [Przybylski A, Gandibleux X (2017)]
- $\epsilon$-constraint method

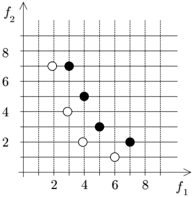
# Basic Notions

Given two arbitrary sets of objective function value vectors in a $Q$-dimensional objective space, $A = \{\boldsymbol{a}^1, \ldots \boldsymbol{a}^m\}$ and $B = \{\boldsymbol{b}^1, \ldots \boldsymbol{b}^n\}$

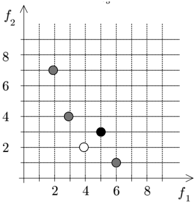
strictly dominates



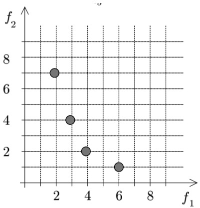

dominates



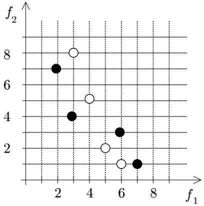

better than




weakly dominates




incomparable



- $z(y)$ the set of points strictly dominating $y$
  $d(y)$ the set of points dominated by $y$:

$$z(y) = \{y' \in \mathbb{R}^p, y' \prec y\}$$
$$d(y) = \{y' \in \mathbb{R}^p, y \leqq y'\}$$

- Let $\boldsymbol{y}^I$ be the ideal point of $Y$, defined by $y^I = \min_{\boldsymbol{y}' \in Y}\{y\}$,

- $\boldsymbol{y}^I$ provides a lower bound on each criterion and can be obtained by solving $p$ programs minimizing each of the $p$ criteria.

- $M$ an upper bound on each criterion, determined by using $p$ max problems over the feasible set.

Finding a nondominated point in a zone is performed by optimizing a strongly monotone function.

**Definition 1.** A function $\phi : Y \to \mathbb{R}$ is *strongly monotone* if and only if for all $(y, y') \in Y^2$, $y \leq y'$ implies $\phi(y) < \phi(y')$.

Standard strongly monotone functions are as follows:

- *The weighted sum*: $\phi(y) = \sum_{i=1}^{p} \lambda_i y_i$ with, $\lambda_i > 0$, $\forall i \in \{1, \ldots, p\}$.
- *Lexicographic*: $\phi(y) = \text{lexmin}\{y_k, \sum_{\substack{i=1 \\ i \neq k}}^{p} \lambda_i y_i\}$, $\lambda_i > 0$, $\forall i \in \{1, \ldots, p\},\ i \neq k$.

The following integer program generates a nondominated point strictly dominating the bound $u$ when a feasible point exists in zone $z(u)$.

**Proposition 1.** *Let $u \in \mathbb{R}^p$ and $\phi : Y \to \mathbb{R}$ be a strongly monotone function. Consider the following program:*

$$P(u)\begin{cases} \min & \phi(y) = \phi(y_1, \dots, y_p) \\ \text{s.t.} & y \in Y, \\ & y_i < u_i \qquad\qquad i \in \{1, \dots, p\}. \end{cases}$$

*Let $y^*$ be an optimal solution of program $P(u)$, if it exists. Then, we have $y^* \in Y_{\mathrm{ND}} \cap z(u)$.*

constraints $y_i < u_i$ cannot be written in a linear program because they involve strict inequalities. We use $y_i \leq u_i - \epsilon_i$ instead, with $\epsilon_i$ smaller than the smallest difference between the performance of two different points on criterion $i$.

Search region that corresponds to the part of the objective space that may contain nondominated points that have not been generated so far.

**Definition 2.** Given a set $N$ of points, the *search region associated with* $N$, referred to as $S(N)$, is defined by

$$S(N) = \mathbb{R}^p \setminus \bigcup_{y \in N} d(y) = \{y \in \mathbb{R}^p : \forall y' \in N, y' \not\leq y\}.$$

Although $N$ usually consists of nondominated points previously generated, it might include any point not even feasible. Observe also that for any point $y \in N$, we have $y \notin S(N)$. Thus, by definition, the search region excludes points already generated.

Generic algorithms iteratively

- explore the search region, or a superset of this region, using the integer program described in Proposition 1;

- then they update it by removing the part dominated by the optimal solution when it exists or by removing the part that is explored when no feasible solution exists.

These two steps are repeated until the search region does not contain feasible points anymore.

Klamroth et al. (2015) describe the search region using $p$-dimensional subregions called search zones, each zone being described using local upper bounds on each objective.

$$S(N) = \Cap_{u \in U(N)} z(u)$$

where $U(N)$ denotes the set of the upper bounds delimiting the search region induced by $N$

When a new nondominated point $y \in S(N) \cap Y_{\mathrm{ND}}$ is found, it is necessary to update the search region by removing the points dominated by $y$. For this purpose, each search zone $z(u) \subseteq S(N)$ containing $y$—such that $y < u$—must be subdivided into $p$ new search zones described by the $p$ following local upper bound: $u^1 = (y_1, u_2, \ldots, u_p)$, $u^2 = (u_1, y_2, u_3, \ldots, u_p)$, and $\ldots$, $u^p = (u_1, \ldots, u_{p-1}, y_p)$. Bound $u^k$ is called the $k^{th}$ child of $u$, $k \in \{1, \ldots, p\}$.

# Defining Points

Proposition 2. Bound $u$ belongs to $U(N)$ if and only if, for any of its bounded component $u_k \neq M$, there exists $y \in N$ such that $y_k = u_k$ and $y_{-k} \prec u_{-k}$.

Let $u \in U(N)$. Points $y \in N$ satisfying Proposition 2 on a given objective $k$ such that

$$\begin{aligned} &y_k = u_k, \\ &y_{-k} \prec u_{-k} \end{aligned}$$

are referred to as the $k$th defining points of bound $u$.