Metaheuristic Optimization

15. Multi-Objective Optimization

Thomas Weise · 汤卫思

tweise@hfuu.edu.cn · http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Faculty of Computer Science and Technology
Institute of Applied Optimization
230601 Shushan District, Hefei, Anhui, China
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99

合肥学院 南艳湖校区/南2区
计算机科学与技术系
应用优化研究所
中国 安徽省 合肥市 蜀山区 230601
经济技术开发区 锦绣大道99号

Outline

website

Section Outline

Definition (Objective Function)

An objective function $f : \mathbb{X} \mapsto \mathbb{R}$ is a (mathematical) function which is subject to optimization.

Definition (Objective Function)

An objective function $f : \mathbb{X} \mapsto \mathbb{R}$ is a (mathematical) function which is subject to optimization.

- Usually subject to minimization

Definition (Objective Function)

An objective function $f : \mathbb{X} \mapsto \mathbb{R}$ is a (mathematical) function which is subject to optimization.

- Usually subject to minimization
- Not necessary a function as we know it from high school (like $f(x) = x^2 + \dots$) but may be arbitrarily complex...

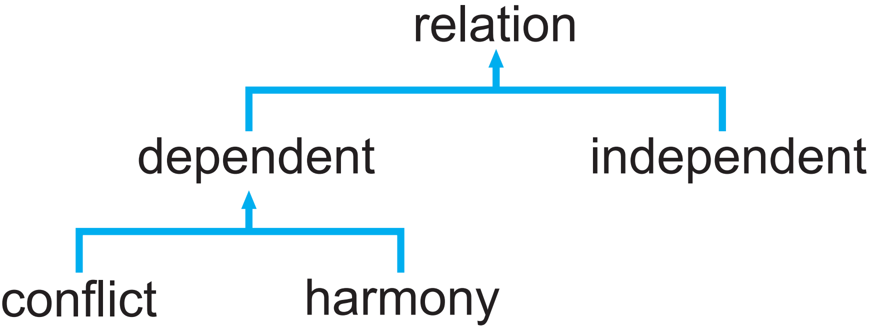
Definition (Multi-Objective Optimization Problem)

In a multi-objective optimization problem (MOP), a set $\vec{f} : \mathbb{X} \mapsto \mathbb{R}^n$ consisting of n objective functions $f_i : \mathbb{X} \mapsto \mathbb{R}$ is to be optimized over a solution space $\mathbb{X}$ [1–3].

$$\vec{f} = \{f_i : \mathbb{X} \mapsto \mathbb{R} : i \in 1 \dots n\} \quad (1)$$

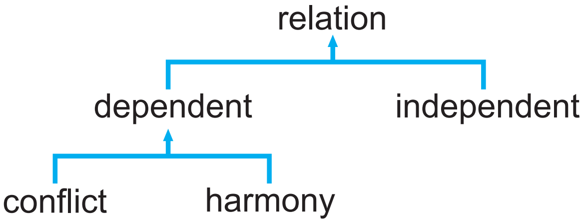
$$\vec{f}(x) = (f_1(x), f_2(x), \dots)^T \implies \vec{f} : \mathbb{X} \mapsto \mathbb{R}^n \quad (2)$$

- These objective functions can have different relations with each other. [4, 5]



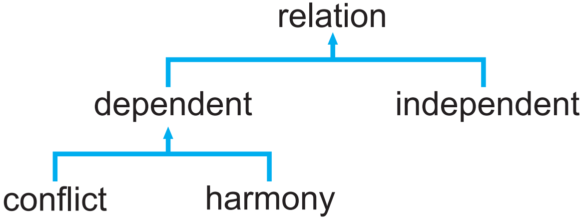
Independent Objectives

- Independent objective functions are unrelated to each other.

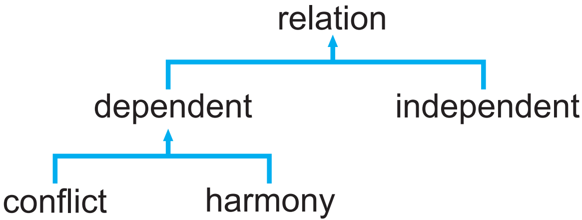


Independent Objectives

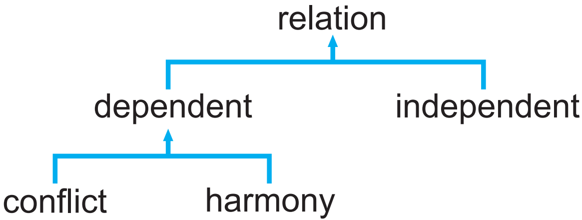
- Independent objective functions are unrelated to each other.
- Example: Find a (1) fast car with (2) beautiful color.



- Independent objective functions are unrelated to each other.
- Example: Find a (1) fast car with (2) beautiful color. $\longrightarrow$ color and speed may be optimized separately

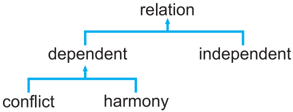


- Independent objective functions are unrelated to each other.
- Example: Find a (1) fast car with (2) beautiful color. $\longrightarrow$ color and speed may be optimized separately
- Uninteresting. Problem can be decomposed into sub-problems which can be optimized separately and solutions of sub-problems can be composed to solution of overall problem

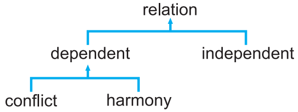


Harmonizing Objectives

- If two functions harmonize, then achieving an improvement in one objective will also lead to an improvement in the other.

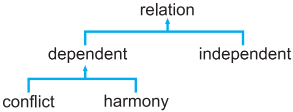


- If two functions harmonize, then achieving an improvement in one objective will also lead to an improvement in the other.
- Example: Find a (1) environmentally friendly car with (2) low fuel consumption.



Harmonizing Objectives

- If two functions harmonize, then achieving an improvement in one objective will also lead to an improvement in the other.
- Example: Find a (1) environmentally friendly car with (2) low fuel consumption. $\longrightarrow$ considering one objective is sufficient

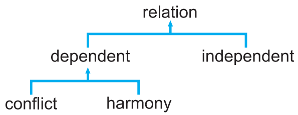


Harmonizing Objectives

- If two functions harmonize, then achieving an improvement in one objective will also lead to an improvement in the other.
- Example: Find a (1) environmentally friendly car with (2) low fuel consumption. $\longrightarrow$ considering one objective is sufficient

$$f_i \sim_X f_j \Rightarrow [f_i(x_1) < f_i(x_2) \Rightarrow f_j(x_1) < f_j(x_2) \forall x_1, x_2 \in X \subseteq \mathbb{X}] \quad (3)$$

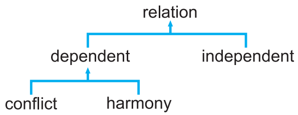
- (definition is given over a subset $X \subseteq \mathbb{X}$ of the solution space)



- If two functions harmonize, then achieving an improvement in one objective will also lead to an improvement in the other.
- Example: Find a (1) environmentally friendly car with (2) low fuel consumption. $\longrightarrow$ considering one objective is sufficient

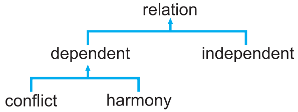
$$f_i \sim_X f_j \Rightarrow [f_i(x_1) < f_i(x_2) \Rightarrow f_j(x_1) < f_j(x_2) \forall x_1, x_2 \in X \subseteq \mathbb{X}] \quad (3)$$

- (definition is given over a subset $X \subseteq \mathbb{X}$ of the solution space)
- Uninteresting. One of the objectives can be omitted / left away, as its presence does neither change the result nor does it make the problem easier

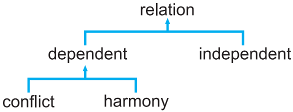


Conflicting Objectives

- If two objectives conflict, then achieving an improvement in one of means getting worse in the other one.



- If two objectives conflict, then achieving an improvement in one of means getting worse in the other one.
- Example: Find a (1) environmentally friendly car which (2) is really fast.

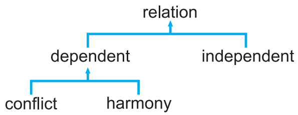


Conflicting Objectives

- If two objectives conflict, then achieving an improvement in one of means getting worse in the other one.
- Example: Find a (1) environmentally friendly car which (2) is really fast.

$$f_i \nsim_X f_j \Rightarrow [f_i(x_1) < f_i(x_2) \Rightarrow f_j(x_1) > f_j(x_2) \forall x_1, x_2 \in X \subseteq \mathbb{X}] \quad (4)$$

- (definition is given over a subset $X \subseteq \mathbb{X}$ of the solution space)

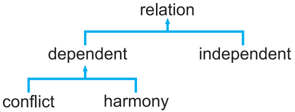


Conflicting Objectives

- If two objectives conflict, then achieving an improvement in one of means getting worse in the other one.
- Example: Find a (1) environmentally friendly car which (2) is really fast.

$$f_i \nsim_X f_j \Rightarrow [f_i(x_1) < f_i(x_2) \Rightarrow f_j(x_1) > f_j(x_2) \forall x_1, x_2 \in X \subseteq \mathbb{X}] \quad (4)$$

- (definition is given over a subset $X \subseteq \mathbb{X}$ of the solution space)
- This is the really interesting situation – here we need to do something!



Objectives

- Usually, objective functions are neither purely harmonizing nor purely conflicting

Objectives

- Usually, objective functions are neither purely harmonizing nor purely conflicting
- Instead, they may harmonize in some parts of the solution space and conflict in others

Optimal?

Before we look any deeper on how to solve multi-objective optimization problems, we should ask ourselfs. . .

Before we look any deeper on how to solve multi-objective optimization problems, we should ask ourselfs...

What does "optimal" mean in the presence of multiple optimization criteria?

Section Outline

Optimization with Priorities

- Give priorities to the different objective functions. [6–9]

Optimization with Priorities

- Give priorities to the different objective functions. [6–9]
- Idea:
  - e.g., speed is more important than environment friendlyness...

- Give priorities to the different objective functions. [6–9]
- Idea:
  - e.g., speed is more important than environment friendlyness...
  - e.g., nice design is more important than functionality...

Optimization with Priorities

- Give priorities to the different objective functions. [6–9]
- Idea:
  - e.g., speed is more important than environment friendlyness...
  - e.g., nice design is more important than functionality...
  - in an n-objective problem, f_1 is more important than f_2, f_2 is more important than f_3, ..., and f_{n-1} is more important than f_n

Optimization with Priorities

- Give priorities to the different objective functions. [6–9]
- Idea:
  - in an n-objective problem, f_1 is more important than f_2, f_2 is more important than f_3, ..., and f_{n-1} is more important than f_n
- Method:

- Give priorities to the different objective functions. [6–9]
- Idea:
  - in an n-objective problem, f_1 is more important than f_2, f_2 is more important than f_3, ..., and f_{n-1} is more important than f_n
- Method:
  1. First consider only f_1 on $\mathbb{X}$ and obtain the set $X^{\star}_{(1)}$ of solutions for this single-objective problem.

- Give priorities to the different objective functions. [6–9]
- Idea:
  - in an n-objective problem, f_1 is more important than f_2, f_2 is more important than f_3, ..., and f_{n-1} is more important than f_n
- Method:
  1. First consider only f_1 on $\mathbb{X}$ and obtain the set $X^{\star}_{(1)}$ of solutions for this single-objective problem.
  2. If $|X^{\star}_{(1)}| > 1$, then consider only f_2 and solve this single-objective problem on $X^{\star}_{(1)}$ and obtain $X^{\star}_{(1,2)}$

- Give priorities to the different objective functions. [6–9]
- Idea:
  - in an n-objective problem, f_1 is more important than f_2, f_2 is more important than f_3, ..., and f_{n-1} is more important than f_n
- Method:
  1. First consider only f_1 on $\mathbb{X}$ and obtain the set $X^{\star}_{(1)}$ of solutions for this single-objective problem.
  2. If $|X^{\star}_{(1)}| > 1$, then consider only f_2 and solve this single-objective problem on $X^{\star}_{(1)}$ and obtain $X^{\star}_{(1,2)}$
  3. If $|X^{\star}_{(1,2)}| > 1$, then consider only f_3 and solve this single-objective problem on $X^{\star}_{(1,2)}$ and obtain $X^{\star}_{(1,2,3)}$

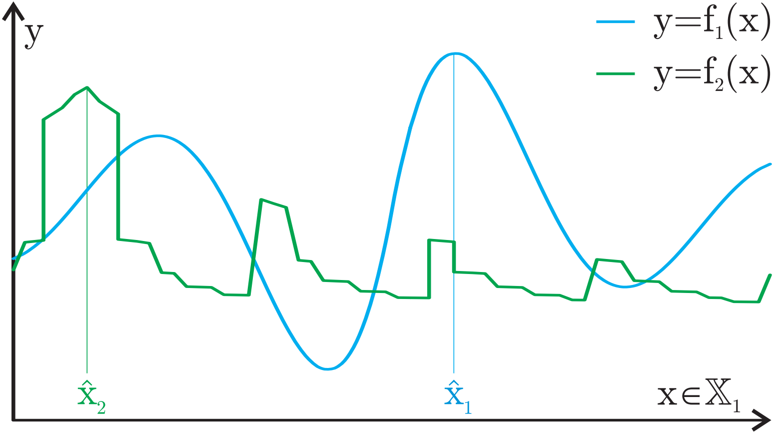
Optimization with Priorities

- Give priorities to the different objective functions. [6–9]
- Idea:
  - in an n-objective problem, f_1 is more important than f_2, f_2 is more important than f_3, ..., and f_{n-1} is more important than f_n
- Method:
  1. First consider only f_1 on $\mathbb{X}$ and obtain the set $X^{\star}_{(1)}$ of solutions for this single-objective problem.
  2. If $|X^{\star}_{(1)}| > 1$, then consider only f_2 and solve this single-objective problem on $X^{\star}_{(1)}$ and obtain $X^{\star}_{(1,2)}$
  3. If $|X^{\star}_{(1,2)}| > 1$, then consider only f_3 and solve this single-objective problem on $X^{\star}_{(1,2)}$ and obtain $X^{\star}_{(1,2,3)}$
  4. and so on...

Optimization with Priorities

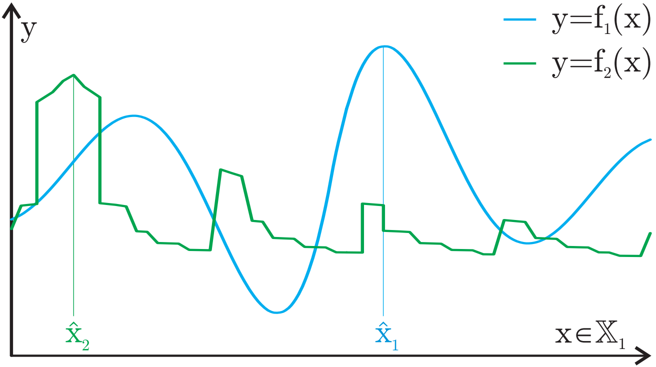
- Give priorities to the different objective functions. [6–9]
- Idea:
  - in an n-objective problem, f_1 is more important than f_2, f_2 is more important than f_3, ..., and f_{n-1} is more important than f_n
- Method:
  1. First consider only f_1 on $\mathbb{X}$ and obtain the set $X^{\star}_{(1)}$ of solutions for this single-objective problem.
  2. If $|X^{\star}_{(1)}| > 1$, then consider only f_2 and solve this single-objective problem on $X^{\star}_{(1)}$ and obtain $X^{\star}_{(1,2)}$
  3. If $|X^{\star}_{(1,2)}| > 1$, then consider only f_3 and solve this single-objective problem on $X^{\star}_{(1,2)}$ and obtain $X^{\star}_{(1,2,3)}$
  4. and so on...
- Separate a multi-objective optimization problem into n single-objective ones

- Two 1-dimensional functions subject to maximization:
$\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$



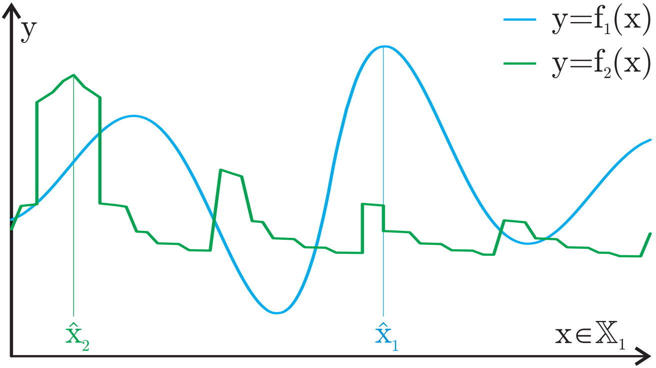
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$



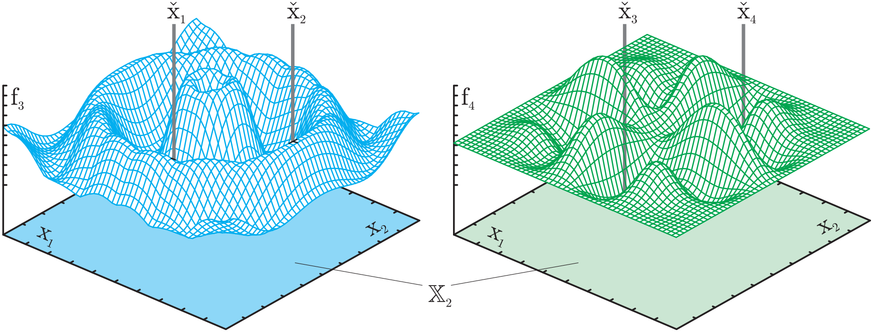
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization: $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- $X^{\star} = X^{\star}_{(1,2)} \cup X^{\star}_{(2,1)} = \{\hat{\hat{x}}_1\} \cup \{\hat{\hat{x}}_2\} = \{\hat{\hat{x}}_1, \hat{\hat{x}}_2\}$

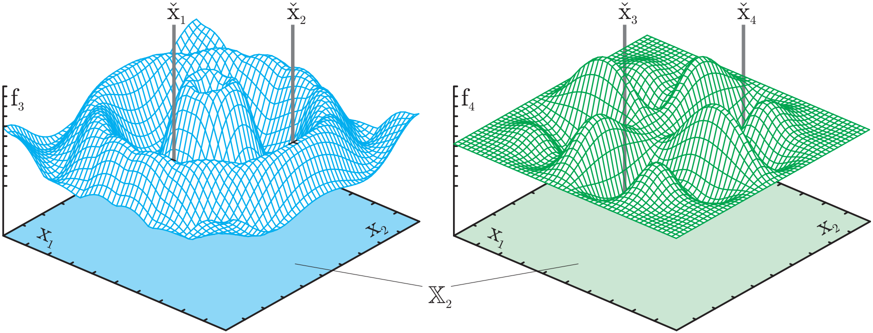


Example B: Two 2d-Functions

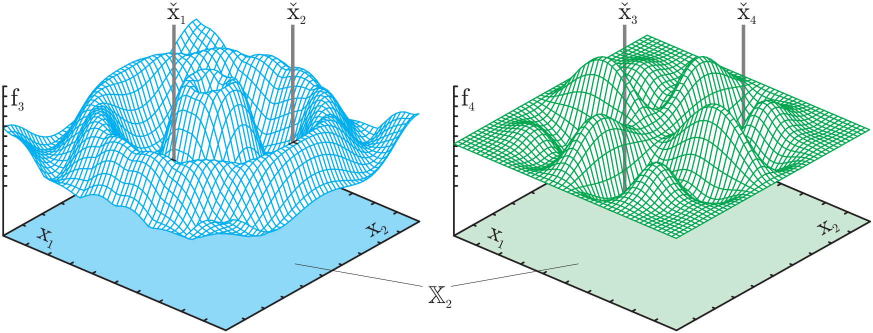
- Two 2-dimensional functions to minimization:
  $\vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R}\,\forall i \in \{3, 4\}$



- Two 2-dimensional functions to minimization: $\vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R}\,\forall i \in \{3,4\}$



- Two 2-dimensional functions to minimization:
  $\vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R} \, \forall i \in \{3, 4\}$
- $X^{\star} = X^{\star}_{(3,4)} \cup X^{\star}_{(4,3)} = \{\check{\check{x}}_1, \check{\check{x}}_2\} \cup \{\check{\check{x}}_3, \check{\check{x}}_4\} = \{\check{\check{x}}_1, \check{\check{x}}_2, \check{\check{x}}_3, \check{\check{x}}_4\}$



Lexicographic Optimization

✔ Easy to perform

- ✔ Easy to perform
- ✘ No trade-off between objectives

- ✔ Easy to perform
- ✘ No trade-off between objectives
- ✘ Only extreme cases will be found

Section Outline

Weighted-Sum Approach

- Sum up the objective values in a weighted sum [10–17]

Weighted-Sum Approach

- Sum up the objective values in a weighted sum [10–17]

$$\mathrm{ws}(x) = \sum_{i=1}^{n} w_i f_i(x) = \sum_{\forall f_i \in \vec{f}} w_i f_i(x) \tag{5}$$

Weighted-Sum Approach

- Sum up the objective values in a weighted sum [10–17]

$$\mathrm{ws}(x) = \sum_{i=1}^{n} w_i f_i(x) = \sum_{\forall f_i \in \vec{f}} w_i f_i(x) \tag{5}$$

- The multi-objective optimization problem is turned into a single-objective problem

Weighted-Sum Approach

- Sum up the objective values in a weighted sum [10–17]

$$\mathrm{ws}(x) = \sum_{i=1}^{n} w_i f_i(x) = \sum_{\forall f_i \in \vec{f}} w_i f_i(x) \tag{5}$$

- The multi-objective optimization problem is turned into a single-objective problem

$$x^{\star} \in X^{\star} \Leftrightarrow \mathrm{ws}(x^{\star}) \leq \mathrm{ws}(x)\ \forall x \in \mathbb{X} \tag{6}$$

- Sum up the objective values in a weighted sum [10–17]

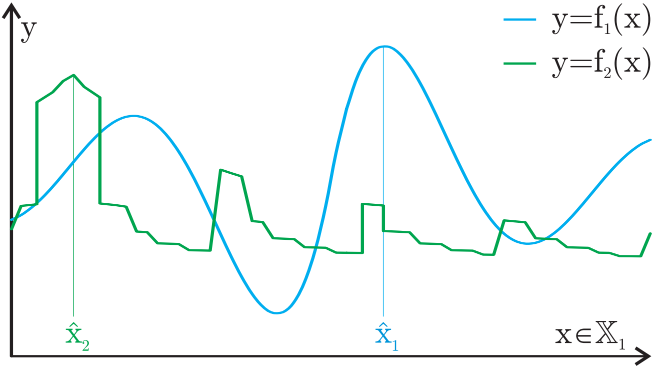
$$\mathrm{ws}(x) = \sum_{i=1}^{n} w_i f_i(x) = \sum_{\forall f_i \in \vec{f}} w_i f_i(x) \tag{5}$$

- The multi-objective optimization problem is turned into a single-objective problem

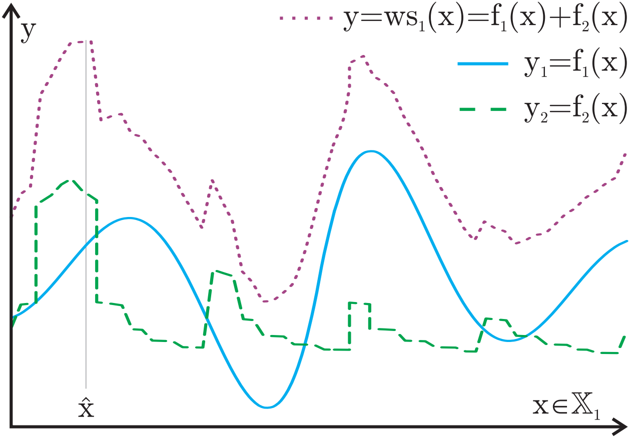
$$x^{\star} \in X^{\star} \Leftrightarrow \mathrm{ws}(x^{\star}) \leq \mathrm{ws}(x)\ \forall x \in \mathbb{X} \tag{6}$$

- We can use any of the single-objective techniques we already know to solve it...

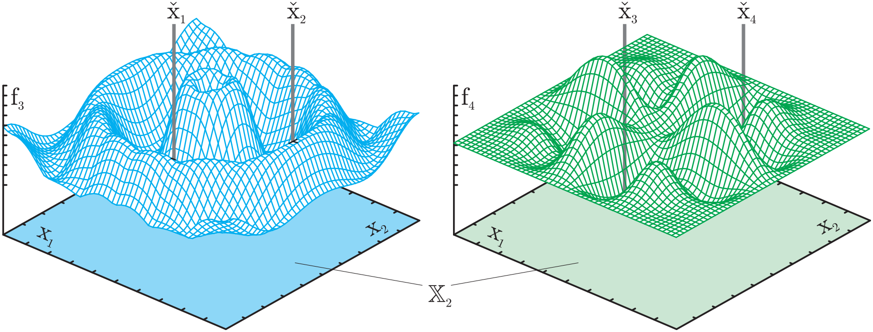
- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R} \, \forall i \in \{1, 2\}$



- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$

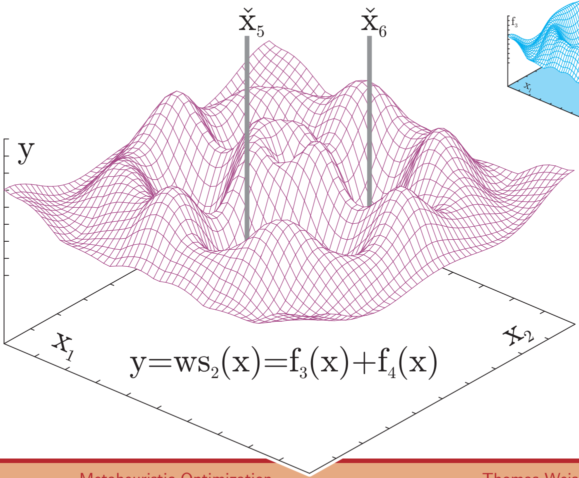


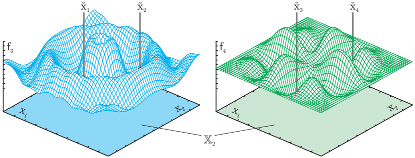
- Two 2-dimensional functions to minimization:
  $\vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R}\,\forall i \in \{3, 4\}$



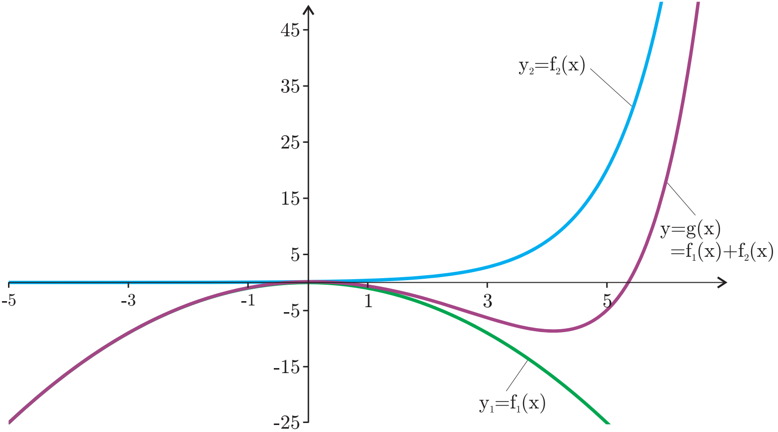
Example B: Two 2d-Functions

- Two 2-dimensional functions to minimization: $\vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R}\, \forall i \in \{3, 4\}$





- Section of two functions subject to either maximization or minimization



✔ Easy to implement

Weighted-Sum Approach

- ✔ Easy to implement
- ✘ Cannot handle objective functions which rise or fall with different speeds properly

- ✔ Easy to implement
- ✘ Cannot handle objective functions which rise or fall with different speeds properly
- ✘ How to set weights properly?

Weighted-Sum Approach

- ✔ Easy to implement
- ✘ Cannot handle objective functions which rise or fall with different speeds properly
- ✘ How to set weights properly?
- ✘ Usually only finds one single element

Weighted-Sum Approach

- ✔ Easy to implement
- ✘ Cannot handle objective functions which rise or fall with different speeds properly
- ✘ How to set weights properly?
- ✘ Usually only finds one single element
- ✘ May not be able to discover the whole trade-off curve

- ✔ Easy to implement
- ✘ Cannot handle objective functions which rise or fall with different speeds properly
- ✘ How to set weights properly?
- ✘ Usually only finds one single element
- ✘ May not be able to discover the whole trade-off curve
- ✘ Objective functions are not always precise measures of utility, adding them up thus does not always make sense

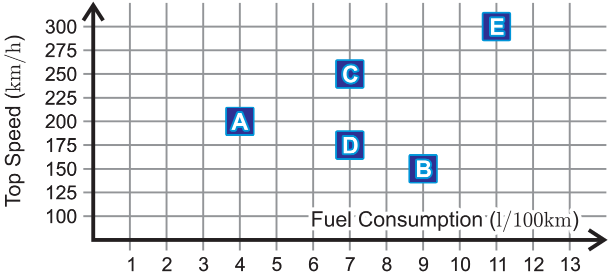
Section Outline

- "Find a fast car which is environmentally friendly!"

Pareto-based Approach

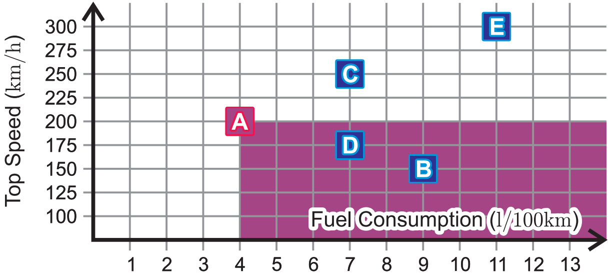
- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km
    Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km
    Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km

- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km

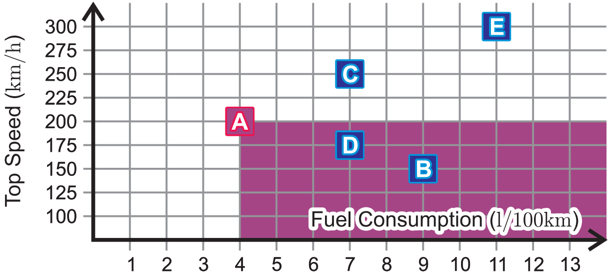


- “Find a fast car which is environmentally friendly!”
  - Assume the following possible candidate solutions:

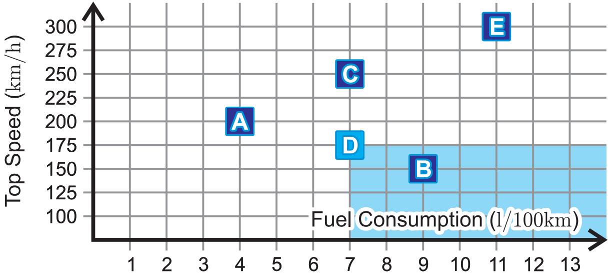
    Car A: 200 km/h with 4 L/100km — Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km — Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - Clearly, car A is better than car D: it is faster and needs less fuels



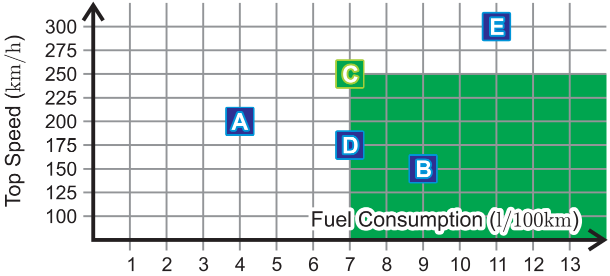
- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - Clearly, car A is better than car D: it is faster and needs less fuels
  - Car A is also better than car B: it is faster and needs less fuel



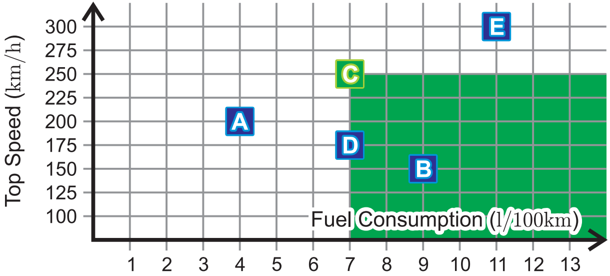
- “Find a fast car which is environmentally friendly!”
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km   Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km   Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - Also, car D is better than car B: it is faster and needs less fuel



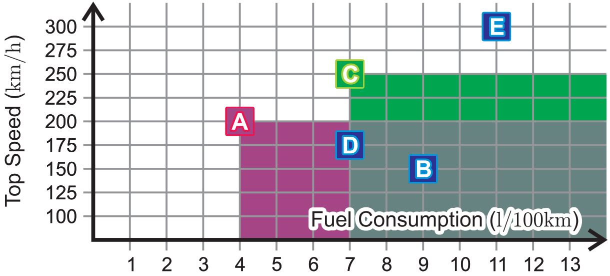
- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - Car C is better than car D: it is faster at the same fuel consumption



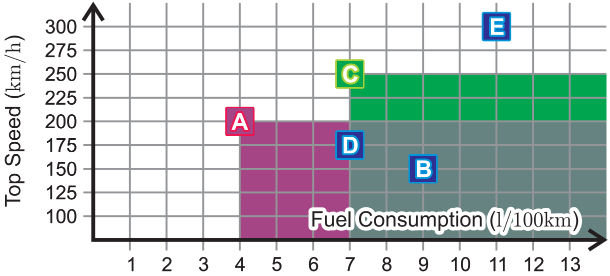
- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km   Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km   Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - Car C is better than car D: it is faster at the same fuel consumption
  - Car C is also better than car B: it is faster and needs less fuel



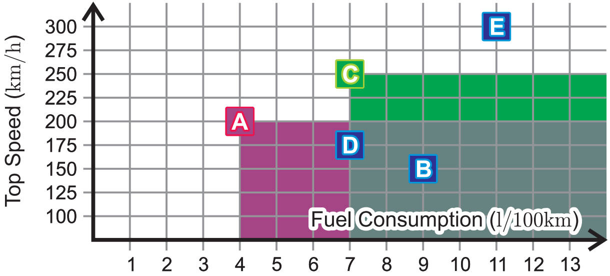
- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km   Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km   Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - However, car C and A cannot be compared!



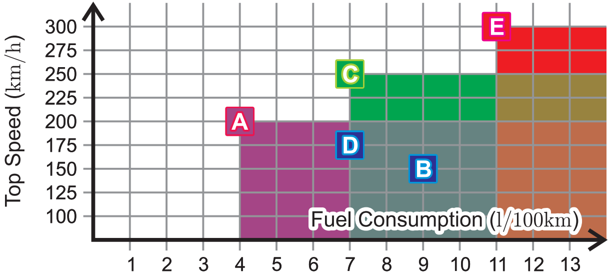
- “Find a fast car which is environmentally friendly!”
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km
    Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km
    Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - However, car C and A cannot be compared!
  - Car A needs less fuel but is slower than car C



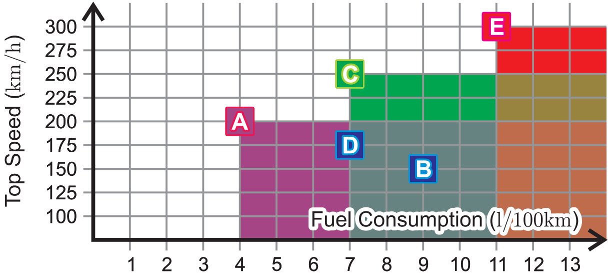
- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km
    Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km
    Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - However, car C and A cannot be compared!
  - Car A needs less fuel but is slower than car C
  - Car C is faster than car A, but needs more fuel



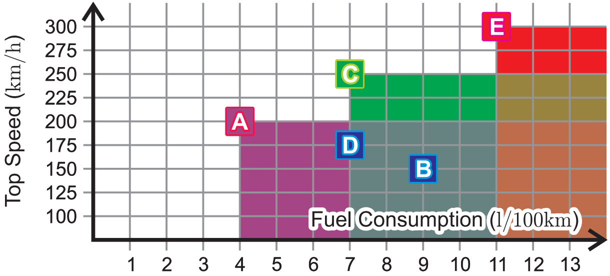
- “Find a fast car which is environmentally friendly!”
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km
    Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km
    Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - No car is better than car E



- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km
    Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km
    Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - No car is better than car E
  - But: No car is worse than car E!



- "Find a fast car which is environmentally friendly!"
  - Assume the following possible candidate solutions:
    Car A: 200 km/h with 4 L/100km Car B: 150 km/h with 9 L/100km
    Car C: 250 km/h with 7 L/100km Car D: 175 km/h with 7 L/100km
    Car E: 300 km/h with 11 L/100km
  - No car is better than car E
  - But: No car is worse than car E!
  - All cars are slower than car E, but all need less fuel



Pareto-based Approach

- Idea first developed by Edgeworth [18] and Pareto [19] in the last two decades of the 19^{th} century

- Idea first developed by Edgeworth [18] and Pareto [19] in the last two decades of the 19^{th} century
- Pareto optimality defines the frontier of solutions that can be reached by trading-off conflicting objectives in an optimal manner [3, 20–25]

- Idea first developed by Edgeworth [18] and Pareto [19] in the last two decades of the 19^{th} century
- Pareto optimality defines the frontier of solutions that can be reached by trading-off conflicting objectives in an optimal manner [3, 20–25]
- Pareto optimality became an important notion in economics, game theory, engineering, and social sciences [26–29].

- Idea first developed in the last two decades of the 19^{th} century[18, 19]
- Pareto optimality defines the frontier of solutions that can be reached by trading-off conflicting objectives in an optimal manner [3, 20–25]

Definition (Domination)

An element x_1 dominates (is preferred to) an element x_2 ($x_1 \dashv x_2$) if x_1 is better than x_2 in at least one objective function and not worse with respect to all other objectives. Based on the set $\vec{f}$ of objective functions f, we can write:

$$x_1 \dashv x_2 \quad \Leftrightarrow \quad \forall i \in 1 \dots n \Rightarrow \omega_i f_i(x_1) \leq \omega_i f_i(x_2) \;\wedge \quad \exists j \in 1 \dots n : \omega_j f_j(x_1) < \omega_j f_j(x_2) \tag{7}$$

$$\omega_i = \begin{cases} 1 & \text{if } f_i \text{ should be minimized} \\ -1 & \text{if } f_i \text{ should be maximized} \end{cases} \tag{8}$$

- Idea first developed in the last two decades of the 19^{th} century[18, 19]
- Pareto optimality defines the frontier of solutions that can be reached by trading-off conflicting objectives in an optimal manner [3, 20–25]

Definition (Pareto Optimal)

An element $x^\star \in \mathbb{X}$ is Pareto optimal (and hence, part of the optimal set $X^\star$) if it is not dominated by any other element in the solution space $\mathbb{X}$. $X^\star$ is called the Pareto-optimal set or Pareto set.

$$x^\star \in X^\star \Leftrightarrow \nexists x \in \mathbb{X} : x \dashv x^\star \tag{7}$$

Pareto-based Approach

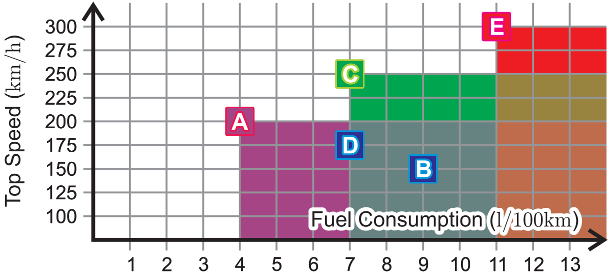
- Idea first developed in the last two decades of the 19^{th} century[18, 19]
- Pareto optimality defines the frontier of solutions that can be reached by trading-off conflicting objectives in an optimal manner [3, 20–25]

Definition (Pareto Frontier)

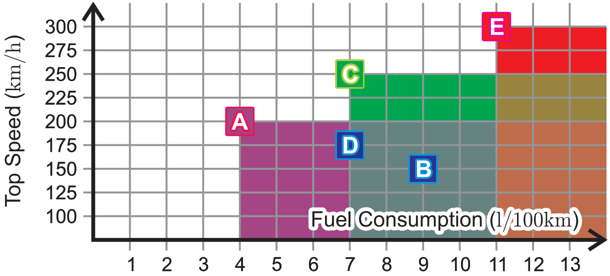
For a given optimization problem, the Pareto front(ier) $F^{\star} \subset \mathbb{R}^n$ is defined as the set of results the objective function vector $\vec{f}$ creates when it is applied to all the elements of the Pareto-optimal set $X^{\star}$.

$$F^{\star} = \{\vec{f}(x^{\star}\} : x^{\star} \in X^{\star}) \tag{7}$$

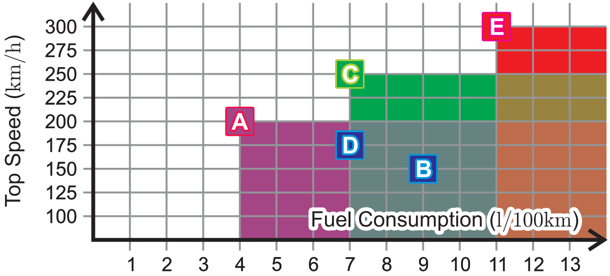
- Non-dominated



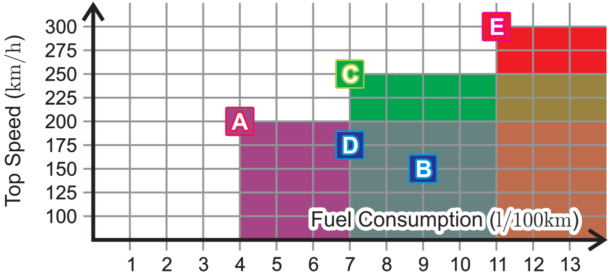
- Non-dominated: A



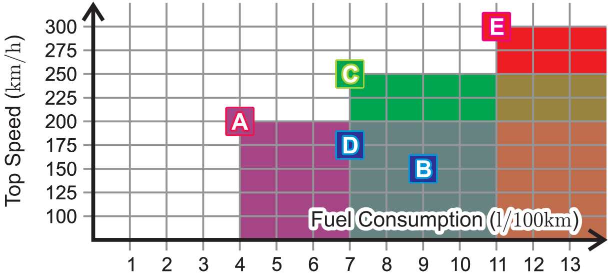
- Non-dominated: A, C



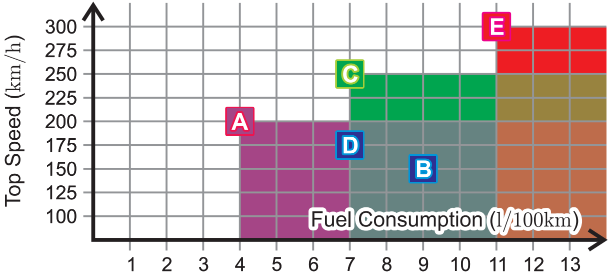
- Non-dominated: A, C, E



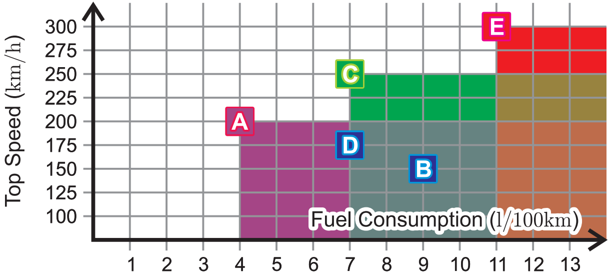
- Non-dominated: A, C, E
- Dominated



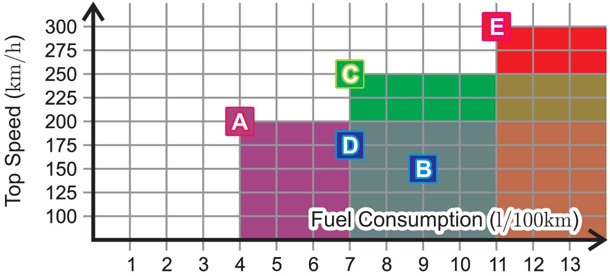
- Non-dominated: A, C, E
- Dominated: D



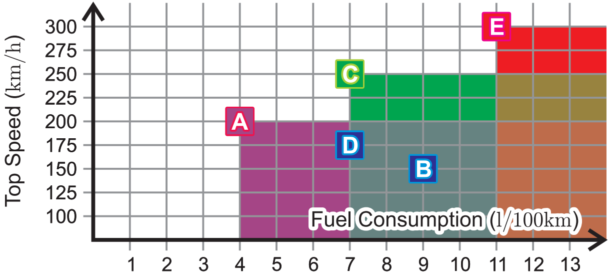
- Non-dominated: A, C, E
- Dominated: D, B



- Non-dominated: A, C, E
- Dominated: D, B
- Pareto set

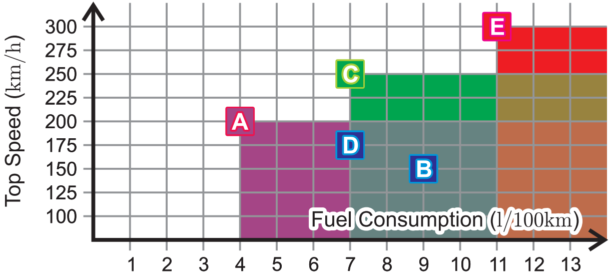


- Non-dominated: A, C, E
- Dominated: D, B
- Pareto set: $X^{\star} = \{A, C, E\}$

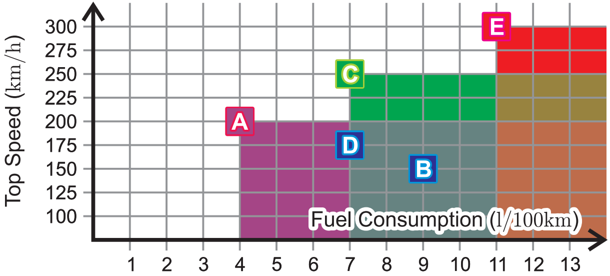


Pareto-based Approach

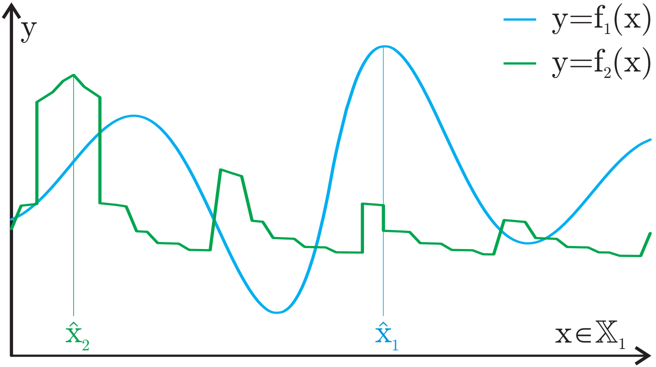
- Non-dominated: A, C, E
- Dominated: D, B
- Pareto set: $X^{\star} = \{A, C, E\}$
- Pareto front



- Non-dominated: A, C, E
- Dominated: D, B
- Pareto set: $X^{\star} = \{A, C, E\}$
- Pareto front: $F^{\star} = \{(4, 200), (7, 250), (11, 300)\}$

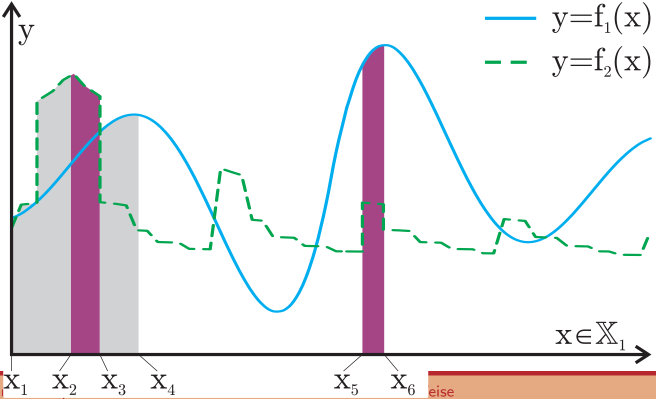


- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R} \, \forall i \in \{1, 2\}$

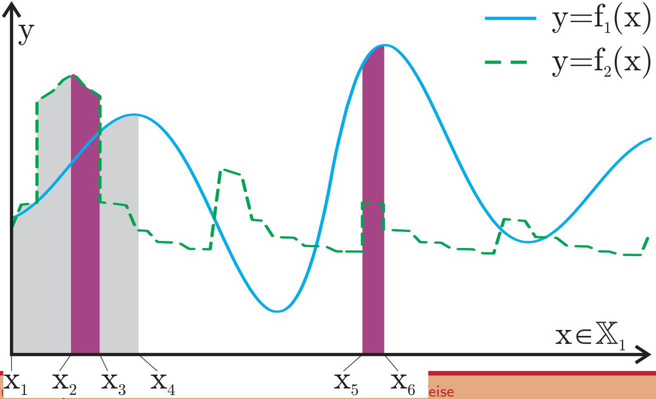


Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- $X^{\star} = [x_2, x_3] \cup [x_5, x_6]$

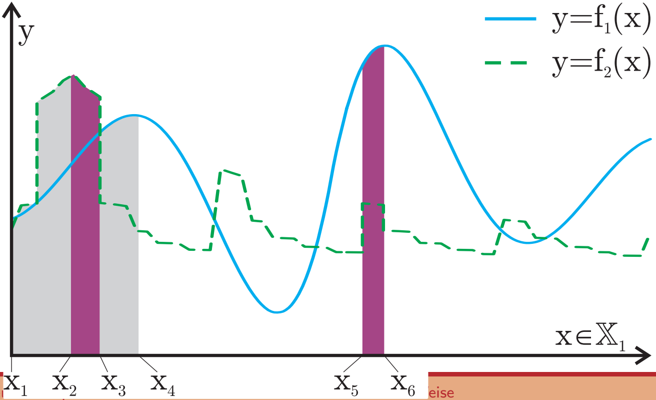


- Two 1-dimensional functions subject to maximization: $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R} \, \forall i \in \{1, 2\}$
- all $x \in [x_1, x_2)$ are dominated by other points in the same region or in $[x_2, x_3]$ – f_1 and f_2 can be improved by increasing x



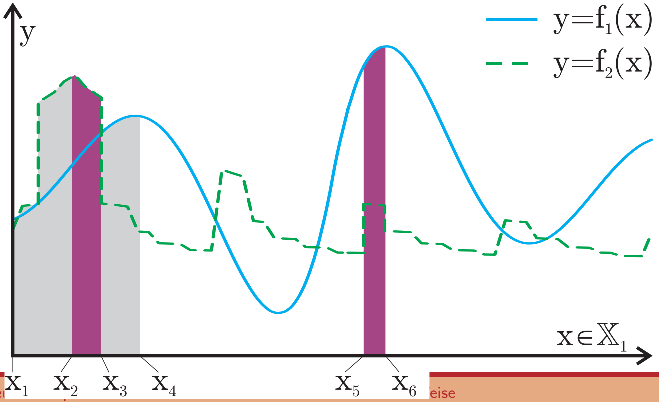
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization: $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R} \, \forall i \in \{1, 2\}$
- f_1 and f_2 *harmonize* in $[x_1, x_2)$: $f_1 \sim_{[x_1,x_2)} f_2$



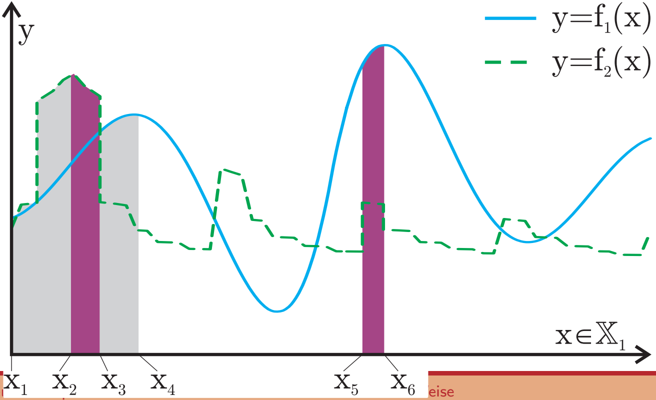
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- $f_1(x_1 + \Delta) > f_1(x_1)$ and $f_2(x_1 + \Delta) > f_2(x_1)$ for all $\Delta \leq x_2 - x_1$



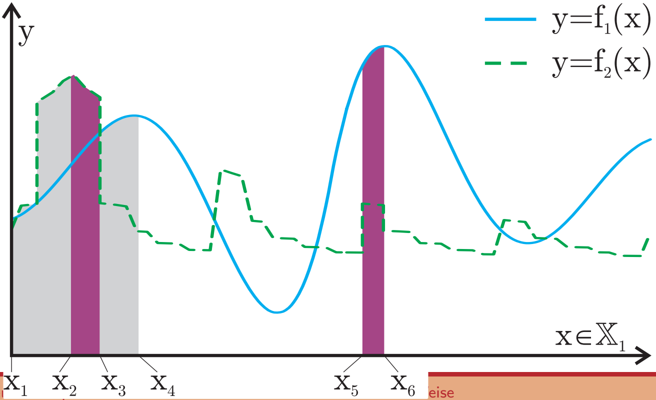
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- If we reach x_2, the situation changes



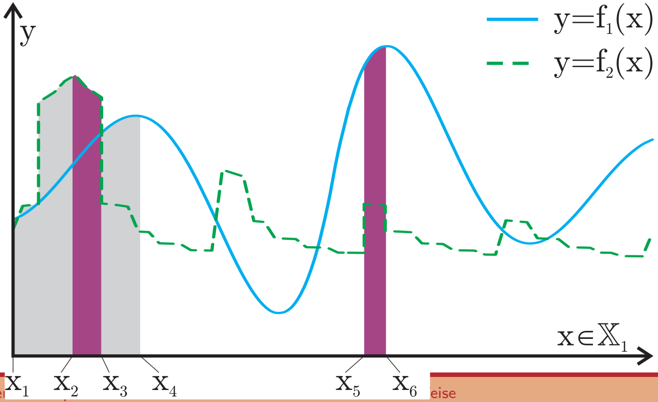
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- x_2 demarks the global maximum of f_2 – the point with the highest possible f_2 value – which can never be dominated

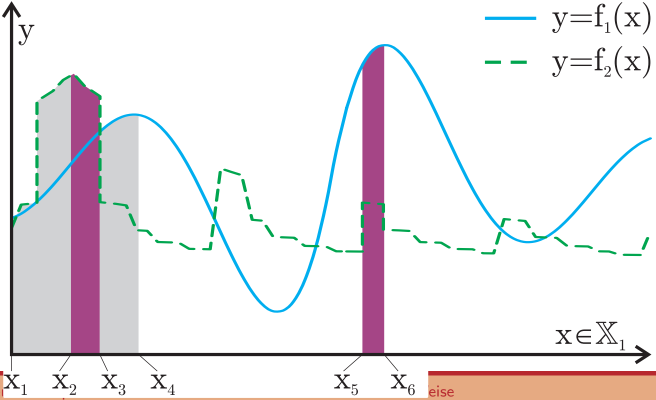


Example A: Two 1d-Functions

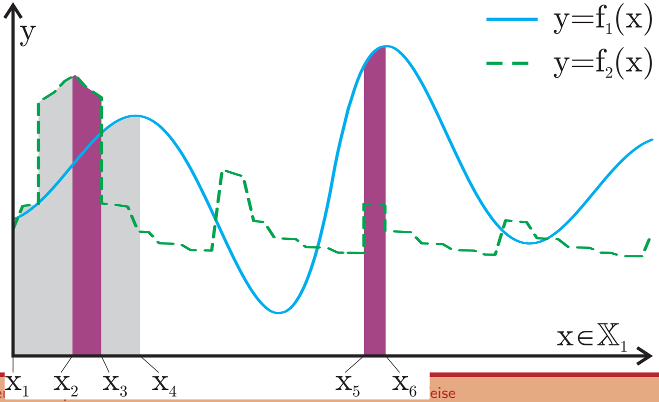
- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R}\,\forall i \in \{1, 2\}$
- From here on, f_2 will decrease for some time, but f_1 keeps rising.



- Two 1-dimensional functions subject to maximization: $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R} \, \forall i \in \{1, 2\}$
- If we now go a small step Δ to the right, we will find a point $x_2 + \Delta$ with $f_2(x_2 + \Delta) < f_2(x_2)$ but also $f_1(x_2 + \Delta) > f_1(x_2)$.

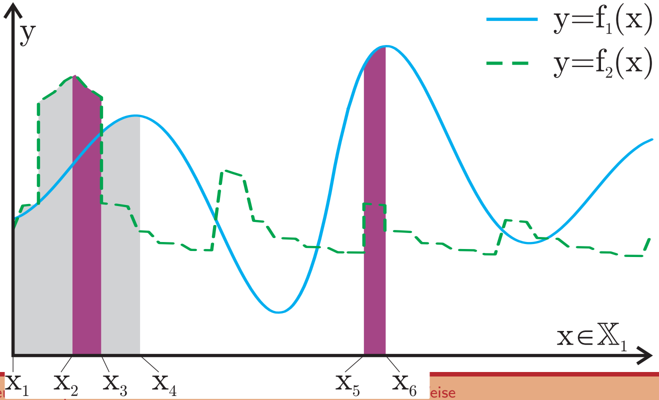


- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- One objective can only get better if another one degenerates, i.e., f_1 and f_2 *conflict* $f_1 \nsim_{[x_2, x_3]} f_2$.



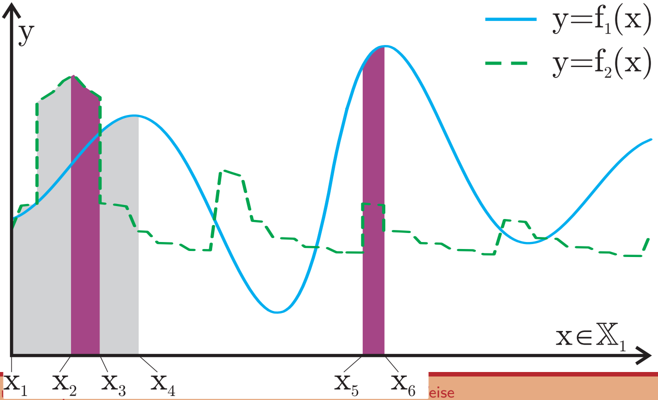
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R}\,\forall i \in \{1, 2\}$
- No point in $[x_1, x_2)$ dominates any point in $[x_2, x_4]$



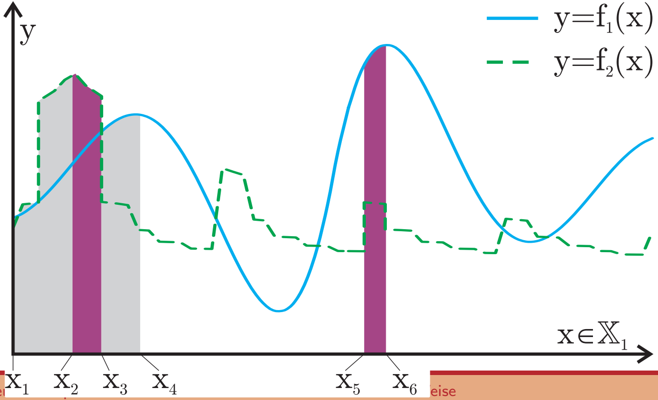
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R} \, \forall i \in \{1, 2\}$
- f_1 keeps rising until x_4 is reached.



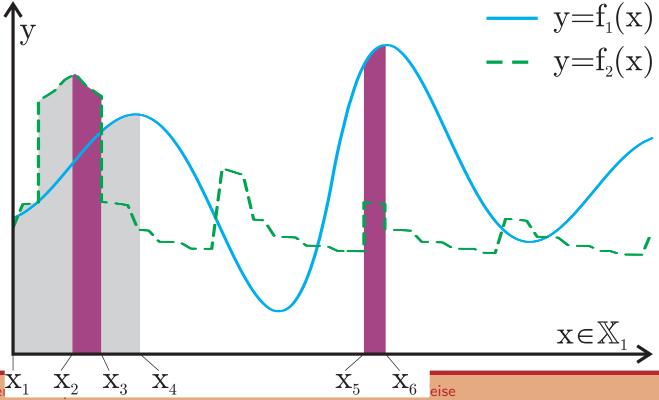
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization: $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- At x_3 however, f_2 steeply falls to a very low level – lower than $f_2(x_5)$.



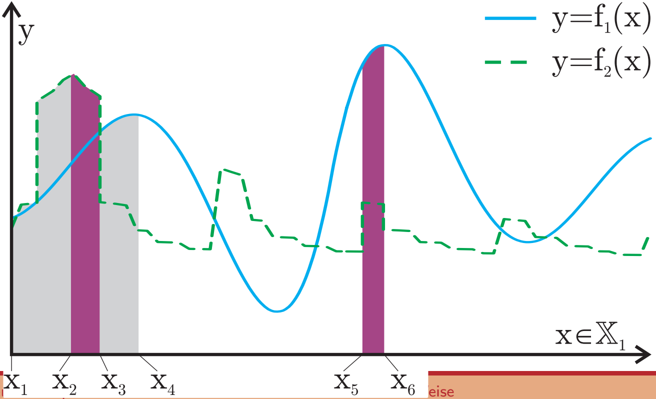
Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- The f_1 values of the points in $[x_5, x_6]$ are also higher than those of the points in $(x_3, x_4]$

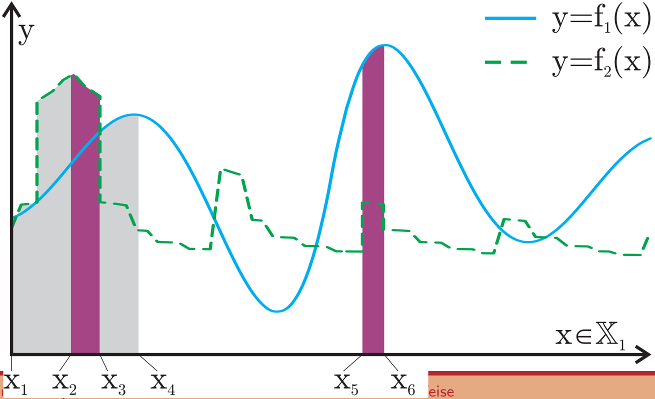


Example A: Two 1d-Functions

- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_{2\}}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1, 2\}$
- All points in the set $[x_5, x_6]$ (which also contains the global maximum of f_1) dominate those in $(x_3, x_4]$.

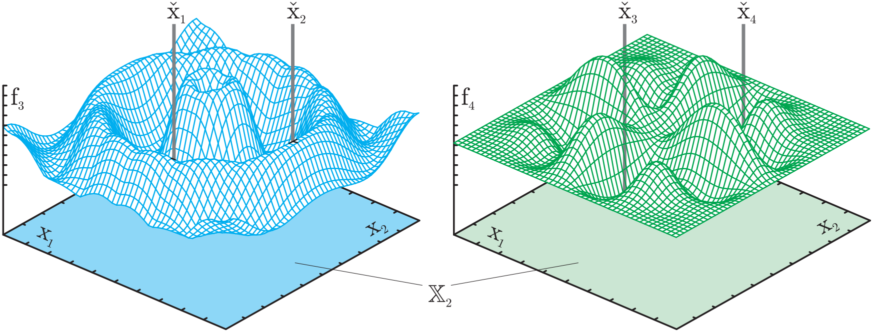


- Two 1-dimensional functions subject to maximization:
  $\vec{f} = \{f_1, f_2\}, f_i : \mathbb{R} \mapsto \mathbb{R}\, \forall i \in \{1,2\}$
- All points in $[x_4, x_5]$ and after x_6 are also dominated by the non-dominated regions just discussed.



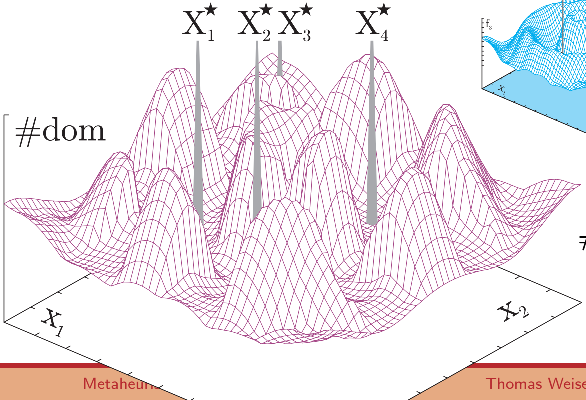
Example B: Two 2d-Functions

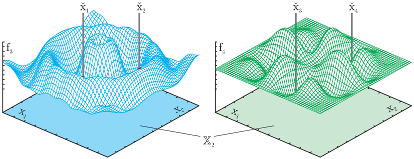
- Grid-based resolution of two 2-dimensional functions to minimization: $\vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R}\, \forall i \in \{3, 4\}$



Example B: Two 2d-Functions

- Grid-based resolution of two 2-dimensional functions to minimization: $\vec{f} = \{f_3, f_4\}, f_i : \mathbb{R}^2 \mapsto \mathbb{R}\,\forall i \in \{3,4\}$
- $X^{\star} = X_1^{\star} \cup X_2^{\star} \cup X_3^{\star} \cup X_4^{\star}$ are not dominated by any other candidate solution

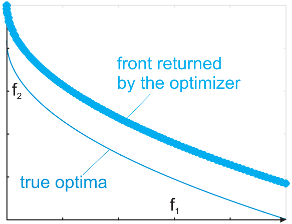




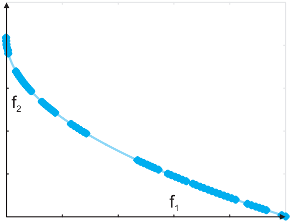
$$\mathbf{\#dom}(\mathbf{x}, \mathbf{X}) = |\{\mathbf{x}' : (x' \in X) \wedge (x' \dashv x)\}|$$

Pareto-based Optimization Results

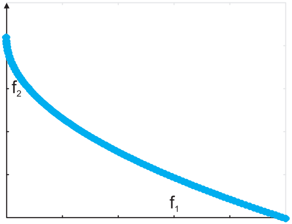
- Goal: Uniformity of convergence – many solutions close to Pareto Front that cover many different "optimal" characteristics



Bad convergence, good spread (uniformity) [30]



Good convergence, bad spread (non-uniformity) [30]



Good convergence, good spread (uniformity) [30]

- ✔ Relative rising/falling speed of objective functions plays no role (big-$\mathcal{O}$ class irrelevant)

Pareto-based Approach

- ✔ Relative rising/falling speed of objective functions plays no role (big-$\mathcal{O}$ class irrelevant)
- ✔ No weights or additional parameters necessary

Pareto-based Approach

- ✔ Relative rising/falling speed of objective functions plays no role (big-$\mathcal{O}$ class irrelevant)
- ✔ No weights or additional parameters necessary
- ✔ Results are best trade-off solutions $\longrightarrow$ Operator can make informed decision

- ✔ Relative rising/falling speed of objective functions plays no role (big-$\mathcal{O}$ class irrelevant)
- ✔ No weights or additional parameters necessary
- ✔ Results are best trade-off solutions $\longrightarrow$ Operator can make informed decision
- ✔ Multiple solutions can be discovered

Pareto-based Approach

- ✔ Relative rising/falling speed of objective functions plays no role (big-$\mathcal{O}$ class irrelevant)
- ✔ No weights or additional parameters necessary
- ✔ Results are best trade-off solutions $\longrightarrow$ Operator can make informed decision
- ✔ Multiple solutions can be discovered
- ✘ Maybe too many solutions will be discovered

- ✔ Relative rising/falling speed of objective functions plays no role (big-$\mathcal{O}$ class irrelevant)
- ✔ No weights or additional parameters necessary
- ✔ Results are best trade-off solutions $\longrightarrow$ Operator can make informed decision
- ✔ Multiple solutions can be discovered
- ✘ Maybe too many solutions will be discovered
- ✘ In many problems, the number of Pareto-optimal solutions may be infinite $\longrightarrow$ Which to chose?

Section Outline

- Multi-objective Evolutionary Algorithms (MOEAs) can deal with multiple, conflicting objectives [1, 2, 7, 31–53]

- Multi-objective Evolutionary Algorithms (MOEAs) can deal with multiple, conflicting objectives [1, 2, 7, 31–53]
- The goal is to return a good approximation of the Pareto front with a good spread

- Multi-objective Evolutionary Algorithms (MOEAs) can deal with multiple, conflicting objectives [1, 2, 7, 31–53]
- The goal is to return a good approximation of the Pareto front with a good spread
- How can we incorporate multiple objectives in the Scheme of EAs?

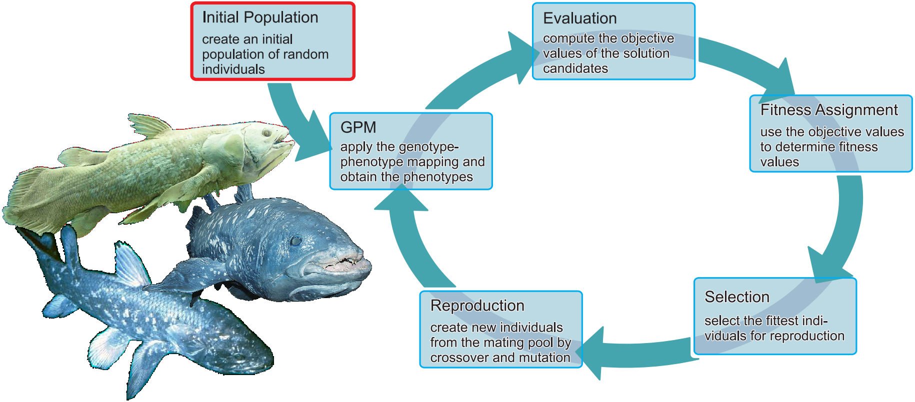
- Multi-objective Evolutionary Algorithms (MOEAs) can deal with multiple, conflicting objectives [1, 2, 7, 31–53]
- The goal is to return a good approximation of the Pareto front with a good spread
- How can we incorporate multiple objectives in the Scheme of EAs?
- Traditional selection schemes cannot be used as is, becaus now we have a vector $\vec{f}(p.x)$ of objective values per individual instead of a single scalar value

- Multi-objective Evolutionary Algorithms (MOEAs) can deal with multiple, conflicting objectives [1, 2, 7, 31–53]
- The goal is to return a good approximation of the Pareto front with a good spread
- How can we incorporate multiple objectives in the Scheme of EAs?
- Traditional selection schemes cannot be used as is, becaus now we have a vector $\vec{f}(p.x)$ of objective values per individual instead of a single scalar value
- Introduce fitness assignment process into the EA which maps the objective value vectors $\vec{f}(p.x)$ to scalar fitness values $\nu(p)$

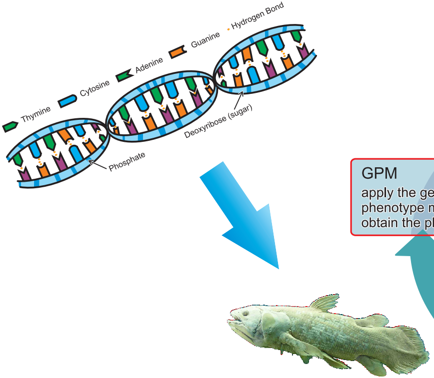
MOEAs and Fitness Assignment

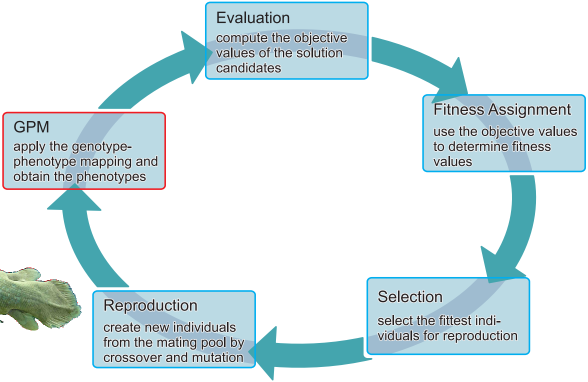
- Multi-objective Evolutionary Algorithms (MOEAs) can deal with multiple, conflicting objectives [1, 2, 7, 31–53]
- The goal is to return a good approximation of the Pareto front with a good spread
- How can we incorporate multiple objectives in the Scheme of EAs?
- Traditional selection schemes cannot be used as is, becaus now we have a vector $\vec{f}(p.x)$ of objective values per individual instead of a single scalar value
- Introduce fitness assignment process into the EA which maps the objective value vectors $\vec{f}(p.x)$ to scalar fitness values $\nu(p)$
- After such a scalar fitness has been assigned, the traditional selection schemes (fitness proportionate, tournament, ...) can be used!

- Nullary search operation to create initial individuals: create a population of random bit strings



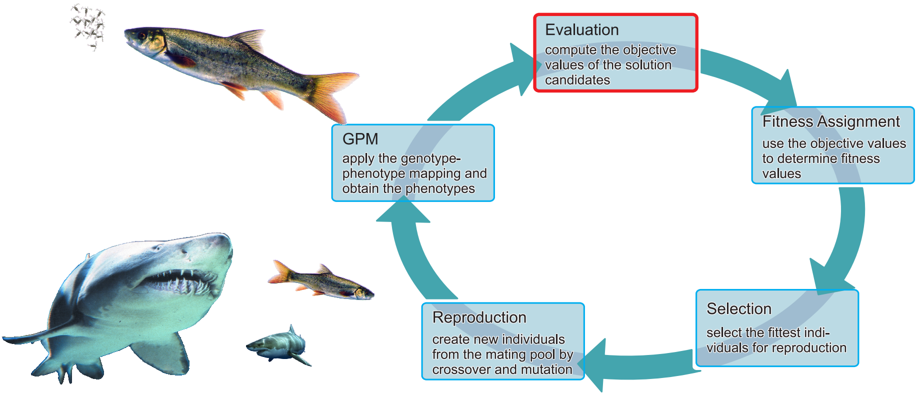
- Map the genotypes to phenotypes
- The GPM is usually problem-dependent



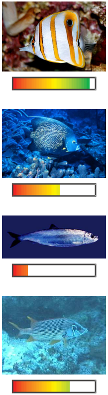


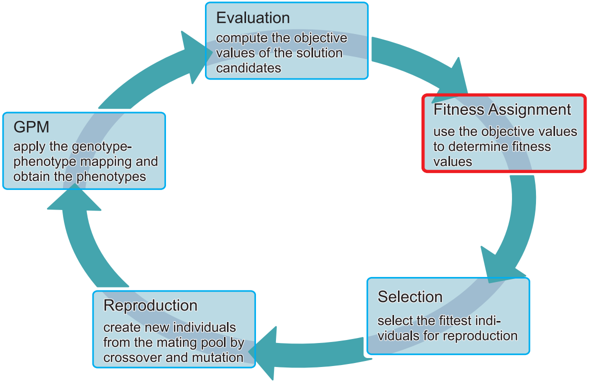
Evolutionary Algorithms: Evaluation

- Evaluate the objective functions – each solution may have different featues
- Each candidate solution x now has a vector of objective values $\vec{f}(x)$

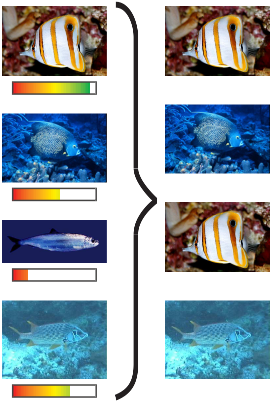


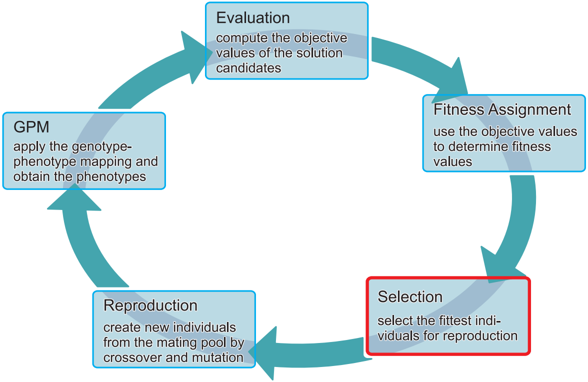
- Fitness is relative: e.g., Pareto optimal *inside population* means non-dominated by the other individuals



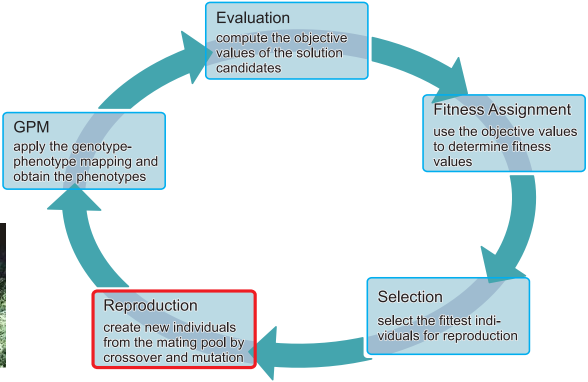


- Select the best individuals with highest probability

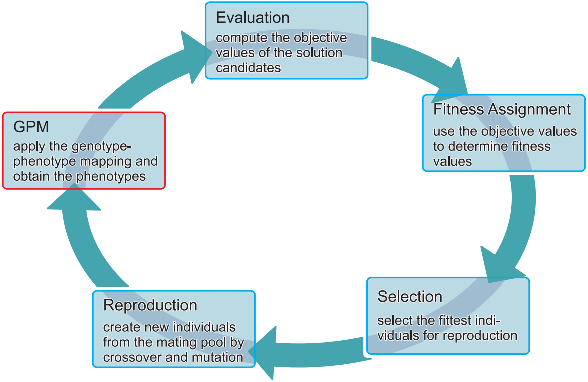




- Mutation and recombination



- Start with new population in next generation.



$\tilde{X} \longleftarrow \text{basicMOEA}(\vec{f}, ps, mps)$

begin

  $\text{pop} \longleftarrow$ create initial population

  while $\neg shouldTerminate$ **do**

    perform genotype-phenotype mapping

    compute objective values

    assign fitness

    $\text{matePool} \longleftarrow$ select parents from pop

    $\text{pop} \longleftarrow$ apply reproduction operations

  return non-dominated solutions in pop

Basic MOEA

$\tilde{X} \longleftarrow \text{basicMOEA}(\vec{f}, ps, mps)$

begin
- $\text{pop} \longleftarrow$ create initial population
- **while** $\neg \mathit{shouldTerminate}$ **do**
  - perform genotype-phenotype mapping
  - compute objective values
  - assign fitness
  - $\text{matePool} \longleftarrow$ select parents from pop
  - $\text{pop} \longleftarrow$ apply reproduction operations
- **return** non-dominated solutions in pop

- A Multi-Objective Evolutionary Algorithm works like a normal EA
- Initialize first generation and generation counter

Basic MOEA

$\tilde{X} \longleftarrow \text{basicMOEA}(\vec{f}, ps, mps)$

begin
- $\text{pop} \longleftarrow$ create initial population
- **while** $\neg shouldTerminate$ **do**
  - perform genotype-phenotype mapping
  - compute objective values
  - assign fitness
  - $\text{matePool} \longleftarrow$ select parents from pop
  - $\text{pop} \longleftarrow$ apply reproduction operations
- **return** non-dominated solutions in pop

- Initialize first generation and generation counter
- Perform genotype-phenotype mapping: transform points in search space to candidate solutions

$\tilde{X} \longleftarrow \mathrm{basicMOEA}(\vec{f}, ps, mps)$

begin
    pop $\longleftarrow$ create initial population
    while $\neg shouldTerminate$ **do**
        perform genotype-phenotype mapping
        compute objective values
        assign fitness
        matePool $\longleftarrow$ select parents from pop
        pop $\longleftarrow$ apply reproduction operations
    return non-dominated solutions in pop

- Perform genotype-phenotype mapping: transform points in search space to candidate solutions
- Compute objective values for all objective functions on all individuals

$\tilde{X} \longleftarrow \text{basicMOEA}(\vec{f}, ps, mps)$

begin
- $\text{pop} \longleftarrow$ create initial population
- **while** $\neg shouldTerminate$ **do**
  - perform genotype-phenotype mapping
  - compute objective values
  - assign fitness
  - $\text{matePool} \longleftarrow$ select parents from pop
  - $\text{pop} \longleftarrow$ apply reproduction operations
- **return** non-dominated solutions in pop

- Compute objective values for all objective functions on all individuals
- Execute fitness assignment process: transform vectors of objective values to scalar fitness

$\tilde{X} \longleftarrow \mathrm{basicMOEA}(\vec{f}, ps, mps)$

begin
  pop $\longleftarrow$ create initial population
  while $\neg shouldTerminate$ **do**
    perform genotype-phenotype mapping
    compute objective values
    assign fitness
    matePool $\longleftarrow$ select parents from pop
    pop $\longleftarrow$ apply reproduction operations
  return non-dominated solutions in pop

- Execute fitness assignment process: transform vectors of objective values to scalar fitness
- Use (e.g., traditional) selection algorithm

$\tilde{X} \longleftarrow \mathrm{basicMOEA}(\vec{f}, ps, mps)$

begin
  $\mathrm{pop} \longleftarrow$ create initial population
  while $\neg shouldTerminate$ **do**
    perform genotype-phenotype mapping
    compute objective values
    assign fitness
    $\mathrm{matePool} \longleftarrow$ select parents from pop
    $\mathrm{pop} \longleftarrow$ apply reproduction operations
  return non-dominated solutions in pop

- Use (e.g., traditional) selection algorithm
- Apply reproduction operators mutation and crossover

Dealing with Multiple Solutions

- So far, we our optimization algorithms return one single solutions

Dealing with Multiple Solutions

- So far, we our optimization algorithms return one single solutions
- Pareto frontier can contain multiple solutions

Dealing with Multiple Solutions

- So far, we our optimization algorithms return one single solutions
- Pareto frontier can contain multiple solutions
- If we want to do Pareto-based optimization, we need to deal with that...

Dealing with Multiple Solutions

- So far, we our optimization algorithms return one single solutions
- Pareto frontier can contain multiple solutions
- If we want to do Pareto-based optimization, we need to deal with that...
- Currently, we update the single, best solution

Dealing with Multiple Solutions

- So far, we our optimization algorithms return one single solutions
- Pareto frontier can contain multiple solutions
- If we want to do Pareto-based optimization, we need to deal with that. . .
- Currently, we update the single, best solution
- Now: Maintain archive of the best (non-dominated) solutions

Dealing with Multiple Solutions

- So far, we our optimization algorithms return one single solutions
- Pareto frontier can contain multiple solutions
- If we want to do Pareto-based optimization, we need to deal with that...
- Currently, we update the single, best solution
- Now: Maintain archive of the best (non-dominated) solutions
- Danger: Archive may grow very big $\Rightarrow$ set size limit

Dealing with Multiple Solutions

- So far, we our optimization algorithms return one single solutions
- Pareto frontier can contain multiple solutions
- If we want to do Pareto-based optimization, we need to deal with that...
- Currently, we update the single, best solution
- Now: Maintain archive of the best (non-dominated) solutions
- Danger: Archive may grow very big $\Rightarrow$ set size limit
- If size limit is reached: delete elements from the archive (maybe randomly, maybe based on density information).

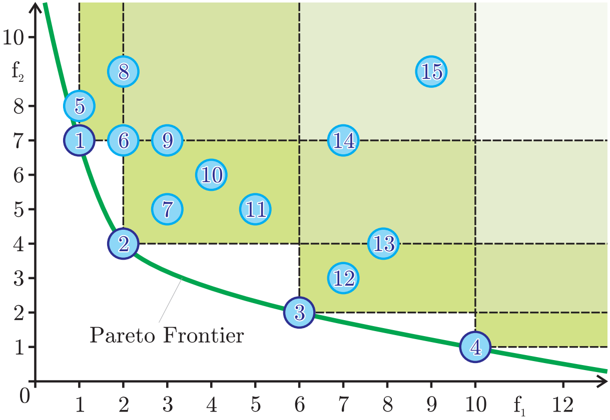
Section Outline

1. Introduction
2. Lexicographic Optimization
3. Weighted-Sum Approach
4. Pareto-based Approach
5. MOEAs
6. Pareto Ranking
7. Problems

Pareto Ranking

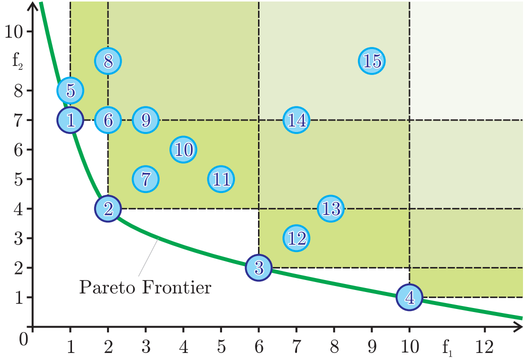
- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]

- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]

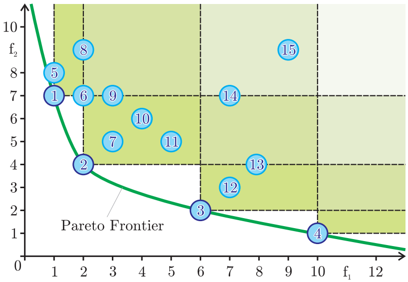


Example: two-objective problem with 15 individuals (1) to (15)

- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Idea: Count the number dominates(p, pop) of individuals that the individual p and set $\nu(p) = {}^{1}/_{(1+\textit{dominates}(p,\mathrm{pop}))}$



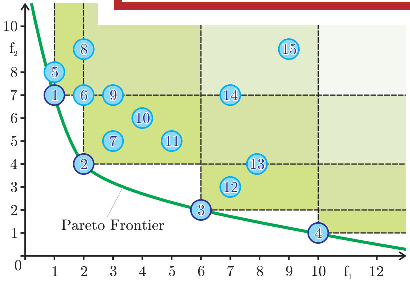
- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Idea: Count the number $\mathrm{dominates}(p, \mathrm{pop})$ of individuals that the individual p and set $\nu(p) = {}^{1}/_{(1+\mathit{dominates}(p,\mathrm{pop}))}$



p	dominated by p	$v(p)$
1	{5, 6, 8, 9, 14, 15}	1/7
2	{6, 7, 8, 9, 10, 11, 13, 14, 15}	1/10
3	{12, 13, 14, 15}	1/5
4	∅	1
5	{8, 15}	1/3
6	{8, 9, 14, 15}	1/5
7	{9, 10, 11, 14, 15}	1/6
8	{15}	1/2
9	{14, 15}	1/3
10	{14, 15}	1/3
11	{14, 15}	1/3
12	{13, 14, 15}	1/4
13	{15}	1/2
14	{15}	1/2
15	∅	1

- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Idea: Count the number $\text{dominates}(p, \text{pop})$ of individuals that the individual p and set $\nu(p)$

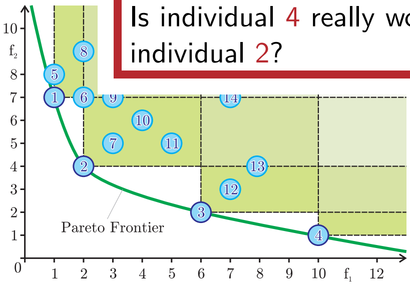
Are individual 4 and 15 really as same as interesting?



p	dominated by p	$v(p)$
1	{5, 6, 8, 9, 14, 15}	1/7
2	{6, 7, 8, 9, 10, 11, 13, 14, 15}	1/10
3	{12, 13, 14, 15}	1/5
4	$\varnothing$	**1**
5	{8, 15}	1/3
6	{8, 9, 14, 15}	1/5
7	{9, 10, 11, 14, 15}	1/6
8	{15}	1/2
9	{14, 15}	1/3
10	{14, 15}	1/3
11	{14, 15}	1/3
12	{13, 14, 15}	1/4
13	{15}	1/2
14	{15}	1/2
15	$\varnothing$	**1**

- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Idea: Count the number $\text{dominates}(p, \text{pop})$ of individuals that the individual p and set $\nu(p)$

Are individual 4 and 15 really as same as interesting?
Is individual 4 really worse than individual 2?



p	dominated by p	$v(p)$
1	{5, 6, 8, 9, 14, 15}	1/7
2	{6, 7, 8, 9, 10, 11, 13, 14, 15}	1/10
3	{12, 13, 14, 15}	1/5
4	$\varnothing$	**1**
5	{8, 15}	1/3
6	{8, 9, 14, 15}	1/5
7	{9, 10, 11, 14, 15}	1/6
8	{15}	1/2
9	{14, 15}	1/3
10	{14, 15}	1/3
11	{14, 15}	1/3
12	{13, 14, 15}	1/4
13	{15}	1/2
14	{15}	1/2
15	$\varnothing$	**1**

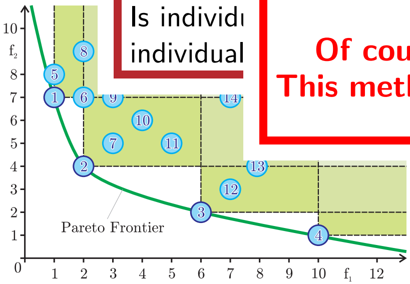
- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Idea: Count the number $\mathrm{dominates}(p, \mathrm{pop})$ of individuals that the individual p and set $\nu(p)$

Are individual 4 and 15 really as same as i

Is individu

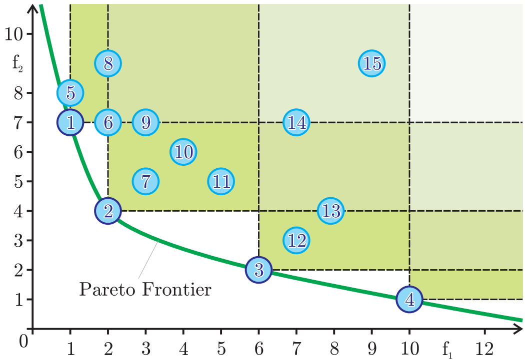
individual

Of course not!
This method is bad.

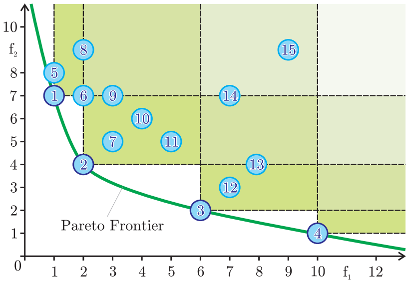


p	dominated by p	$v(p)$
1	{5, 6, 8, 9, 14, 15}	1/7
2	{6, 7, 8, 9, 10, 11, 13, 14, 15}	1/10
3	{12, 13, 14, 15}	1/5
4	∅	**1**
5	8, 15}	1/3
	8, 9, 14, 15}	1/5
	9, 10, 11, 14, 15}	1/6
	15}	1/2
	14, 15}	1/3
	14, 15}	1/3
	14, 15}	1/3
12	{13, 14, 15}	1/4
13	{15}	1/2
14	{15}	1/2
15	∅	**1**

- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Idea: Count the number $\#\mathrm{dom}(p, \mathrm{pop})$ of individuals in the population that dominate individual p, set $\nu(p) = \#\mathrm{dom}(p, \mathrm{pop})$



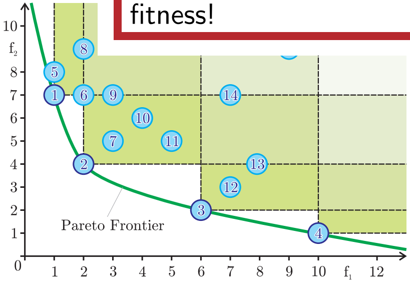
- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Pareto Ranking: Count the number $\#\mathrm{dom}(p, \mathrm{pop})$ of individuals in the population that dominate individual p, set $\nu(p) = \#\mathrm{dom}(p, \mathrm{pop})$



p	#dom(p, Pop)	$v(p)$
1	∅	0
2	∅	0
3	∅	0
4	∅	0
5	{1}	1
6	{1, 2}	2
7	{2}	1
8	{1, 2, 5, 6}	4
9	{1, 2, 6, 7}	4
10	{2, 7}	2
11	{2, 7}	2
12	{3}	1
13	{2, 3, 12}	3
14	{1, 2, 3, 6, 7, 9, 10, 11, 12}	9
15	{1,2,3,5,6,7,8,9,10,11,12,13,14}	13

- Reflect the Pareto dominance relationship of the individuals in the population in the fitness! [41, 54–56]
- Pareto Ranking: Count the number $\#\mathrm{dom}(p, \mathrm{pop})$ of individuals in the population that dominate individual p, se

All the non-dominated individuals have the same, best possible fitness!



p	#dom(p, Pop)	$v(p)$
1	∅	**0**
2	∅	**0**
3	∅	**0**
4	∅	**0**
5	{1}	1
6	{1, 2}	2
7	{2}	1
8	{1, 2, 5, 6}	4
9	{1, 2, 6, 7}	4
10	{2, 7}	2
11	{2, 7}	2
12	{3}	1
13	{2, 3, 12}	3
14	{1, 2, 3, 6, 7, 9, 10, 11, 12}	9
15	{1,2,3,5,6,7,8,9,10,11,12,13,14}	13

Pareto Ranking

paretoRank(pop)

begin
  22**foreach** $p \in pop$ **do** $p.y \longleftarrow 0$
  for $i \longleftarrow 1$ **up to** $ps - 1$ **do**
    $p_1 \longleftarrow \mathrm{pop}[i]$
    for $j \longleftarrow 0$ **up to** $i - 1$ **do**
      $p_2 \longleftarrow \mathrm{pop}[j]$
      if $p_1 \dashv p_2$ **then** $p_2.y \longleftarrow p_2.y + 1$
      if $p_2 \dashv p_1$ **then** $p_1.y \longleftarrow p_1.y + 1$

- Pareto ranking can easily be implemented (see [56] for better method):

Pareto Ranking

paretoRank(pop)

```
begin
    22foreach p ∈ pop do p.y ⟵ 0

    for i ⟵ 1 up to ps − 1 do
        p1 ⟵ pop[i]
        for j ⟵ 0 up to i − 1 do
            p2 ⟵ pop[j]
            if p1 ⊣ p2 then p2.y ⟵ p2.y + 1

            if p2 ⊣ p1 then p1.y ⟵ p1.y + 1
```

- Pareto ranking can easily be implemented (see [56] for better method):
- Initialize fitness of all individuals to zero (Fitness is subject to minimization)

Pareto Ranking

paretoRank(pop)

begin
  22**foreach** $p \in pop$ **do** $p.y \longleftarrow 0$

  for $i \longleftarrow 1$ **up to** $ps - 1$ **do**
    $p_1 \longleftarrow \mathrm{pop}[i]$
    for $j \longleftarrow 0$ **up to** $i - 1$ **do**
      $p_2 \longleftarrow \mathrm{pop}[j]$
      if $p_1 \dashv p_2$ **then** $p_2.y \longleftarrow p_2.y + 1$

      if $p_2 \dashv p_1$ **then** $p_1.y \longleftarrow p_1.y + 1$

- Pareto ranking can easily be implemented (see [56] for better method):
- Initialize fitness of all individuals to zero (Fitness is subject to minimization)
- Compare each individual $p_1 \ldots$

paretoRank(pop)

```
begin
    22foreach p ∈ pop do p.y ⟵ 0

    for i ⟵ 1 up to ps − 1 do
        p₁ ⟵ pop[i]
        for j ⟵ 0 up to i − 1 do
            p₂ ⟵ pop[j]
            if p₁ ⊣ p₂ then p₂.y ⟵ p₂.y + 1

            if p₂ ⊣ p₁ then p₁.y ⟵ p₁.y + 1
```

- Pareto ranking can easily be implemented (see [56] for better method):
- Compare each individual p_1...
- ...with each other individual p_2

paretoRank(pop)

```
begin
    22foreach p ∈ pop do p.y ⟵ 0

    for i ⟵ 1 up to ps − 1 do
        p1 ⟵ pop[i]
        for j ⟵ 0 up to i − 1 do
            p2 ⟵ pop[j]
            if p1 ⊣ p2 then p2.y ⟵ p2.y + 1

            if p2 ⊣ p1 then p1.y ⟵ p1.y + 1
```

- Pareto ranking can easily be implemented (see [56] for better method):
- . . . with each other individual p_2
- If p_1 wins, count it as loss for p_2

paretoRank(pop)

begin
  22**foreach** $p \in pop$ **do** $p.y \longleftarrow 0$

  for $i \longleftarrow 1$ **up to** $ps - 1$ **do**
    $p_1 \longleftarrow \mathrm{pop}[i]$
    for $j \longleftarrow 0$ **up to** $i - 1$ **do**
      $p_2 \longleftarrow \mathrm{pop}[j]$
      if $p_1 \dashv p_2$ **then** $p_2.y \longleftarrow p_2.y + 1$

      if $p_2 \dashv p_1$ **then** $p_1.y \longleftarrow p_1.y + 1$

- Pareto ranking can easily be implemented (see [56] for better method):
- If p_1 wins, count it as loss for p_2
- If p_2 wins, count it as loss for p_1

How to use Pareto Ranking?

- Pareto Ranking translates the vectors of objective values to scalar fitness, depending on the population structure

How to use Pareto Ranking?

- Pareto Ranking translates the vectors of objective values to scalar fitness, depending on the population structure
- Scale of the fitness values independent of scale of objective values

How to use Pareto Ranking?

- Pareto Ranking translates the vectors of objective values to scalar fitness, depending on the population structure
- Scale of the fitness values independent of scale of objective values
- Can be comined with all traditional selection schemes

How to use Pareto Ranking?

- Pareto Ranking translates the vectors of objective values to scalar fitness, depending on the population structure
- Scale of the fitness values independent of scale of objective values
- Can be comined with all traditional selection schemes:
  1. Tournament Selection

How to use Pareto Ranking?

- Pareto Ranking translates the vectors of objective values to scalar fitness, depending on the population structure
- Scale of the fitness values independent of scale of objective values
- Can be comined with all traditional selection schemes:
  1. Tournament Selection
  2. Truncation Selection

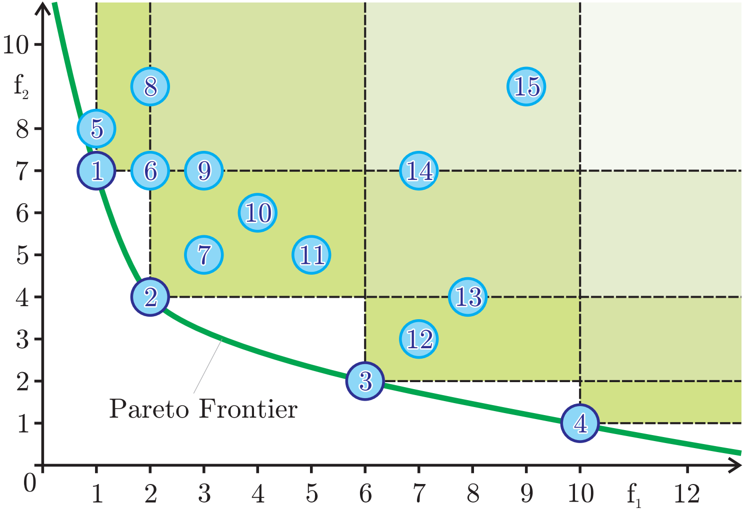
How to use Pareto Ranking?

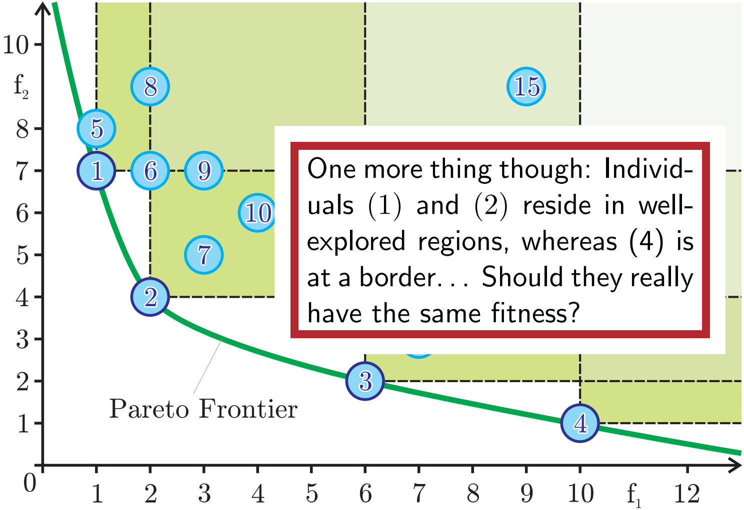
- Pareto Ranking translates the vectors of objective values to scalar fitness, depending on the population structure
- Scale of the fitness values independent of scale of objective values
- Can be comined with all traditional selection schemes:
  1. Tournament Selection
  2. Truncation Selection
  3. also with Roulette Wheel selection

How to use Pareto Ranking?

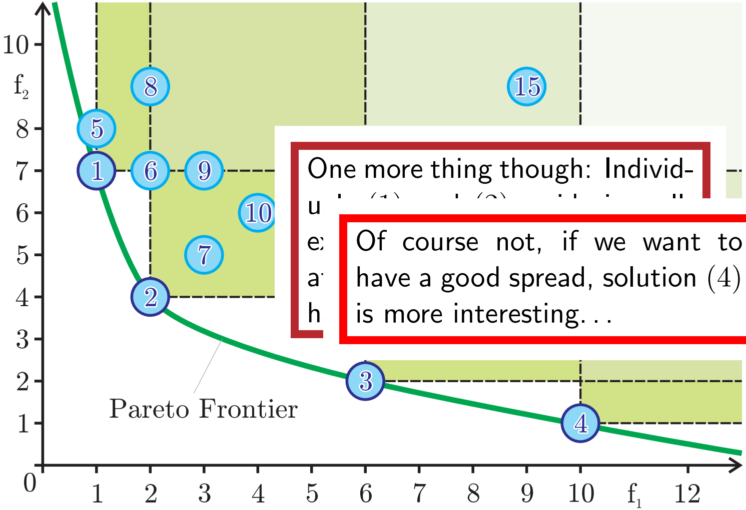
- Pareto Ranking translates the vectors of objective values to scalar fitness, depending on the population structure
- Scale of the fitness values independent of scale of objective values
- Can be comined with all traditional selection schemes:
  1. Tournament Selection
  2. Truncation Selection
  3. also with Roulette Wheel selection: Pareto rank is scale-independent and thus, the problems of Roulette Wheel selection (for fitness minimization) do not occur...

Section Outline


10
f2
8
7
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 f1 12
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Pareto Frontier



One more thing though: Individuals (1) and (2) reside in well-explored regions, whereas (4) is at a border... Should they really have the same fitness?



One more thing though: Individ-
u
e
a
h

Of course not, if we want to have a good spread, solution (4) is more interesting...

Practical MOEAs

- Most practical MOEAs combine Pareto ranking information with some form of *diversity enhancing* measure!

Practical MOEAs

- Most practical MOEAs combine Pareto ranking information with some form of *diversity enhancing* measure! (actually, mose real-world EAs do that. . .)

- Most practical MOEAs combine Pareto ranking information with some form of *diversity enhancing* measure!
- Good examples are:

Practical MOEAs

- Most practical MOEAs combine Pareto ranking information with some form of *diversity enhancing* measure!
- Good examples are:
  1. NSGA-II by Deb et al. [57] (Pareto rank combined with the crowding distance in objective space)

Practical MOEAs

- Most practical MOEAs combine Pareto ranking information with some form of *diversity enhancing* measure!
- Good examples are:
  1. NSGA-II by Deb et al. [57] (Pareto rank combined with the crowding distance in objective space)
  2. SPEA-2 by Zitzler et al. [58] (Pareto-domination based strength together with distance to the k nearest neighbor in objective space)

- Most practical MOEAs combine Pareto ranking information with some form of *diversity enhancing* measure!
- Good examples are:
  1. NSGA-II by Deb et al. [57] (Pareto rank combined with the crowding distance in objective space)
  2. SPEA-2 by Zitzler et al. [58] (Pareto-domination based strength together with distance to the k nearest neighbor in objective space)
  3. PESA by Corne et al. [59] (Pareto domination and number of other individuals in the same hyper-box in a grid defined over the search space)

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion:
  1. set $i \longleftarrow 1$

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion:
  1. set $i \longleftarrow 1$
  2. Find all non-dominated individuals, put them into "front i"

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion:
  1. set $i \longleftarrow 1$
  2. Find all non-dominated individuals, put them into "front i"
  3. (temporarily) remove all individuals in "front i" from population

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion:
  1. set $i \longleftarrow 1$
  2. Find all non-dominated individuals, put them into "front i"
  3. (temporarily) remove all individuals in "front i" from population
  4. Set $i \longleftarrow i + 1$

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion:
  1. set $i \longleftarrow 1$
  2. Find all non-dominated individuals, put them into "front i"
  3. (temporarily) remove all individuals in "front i" from population
  4. Set $i \longleftarrow i + 1$
  5. If population not empty, go to step 1.

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*:
  1. For each individual p set $p.cd \longleftarrow 0$

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*:
  1. For each individual p set $p.cd \longleftarrow 0$
  2. For each objective f_j do

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*:
  1. For each individual p set $p.\mathit{cd} \longleftarrow 0$
  2. For each objective f_j do:
     1. $L_j \longleftarrow$ list of the k individuals in "fromt i" sorted according to f_j

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*:
  1. For each individual p set $p.\mathit{cd} \longleftarrow 0$
  2. For each objective f_j do:
     1. $L_j \longleftarrow$ list of the k individuals in "fromt i" sorted according to f_j
     2. Set $L_j[1].\mathit{cd} \longleftarrow L_j[k].\mathit{cd} \longleftarrow \infty$

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*:
  1. For each individual p set $p.\mathit{cd} \longleftarrow 0$
  2. For each objective f_j do:
     1. $L_j \longleftarrow$ list of the k individuals in "fromt i" sorted according to f_j
     2. Set $L_j[1].\mathit{cd} \longleftarrow L_j[k].\mathit{cd} \longleftarrow \infty$
     3. For $l \longleftarrow 2$ to $\longleftarrow k - 1$ do

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*:
  1. For each individual p set $p.\mathit{cd} \longleftarrow 0$
  2. For each objective f_j do:
     1. $L_j \longleftarrow$ list of the k individuals in "fromt i" sorted according to f_j
     2. Set $L_j[1].\mathit{cd} \longleftarrow L_j[k].\mathit{cd} \longleftarrow \infty$
     3. For $l \longleftarrow 2$ to $\longleftarrow k-1$ do:
        $L_j[l].\mathit{cd} \longleftarrow L_j[l].\mathit{cd} + \frac{(L_j[l+1].f_j - L_j[l-1].f_j)}{L_j[k] - L_j[k-1]}$

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*:
  1. For each individual p set $p.\mathit{cd} \longleftarrow 0$
  2. For each objective f_j do:
     1. $L_j \longleftarrow$ list of the k individuals in "fromt i" sorted according to f_j
     2. Set $L_j[1].\mathit{cd} \longleftarrow L_j[k].\mathit{cd} \longleftarrow \infty$
     3. For $l \longleftarrow 2$ to $\longleftarrow k-1$ do:
        $L_j[l].\mathit{cd} \longleftarrow L_j[l].\mathit{cd} + \frac{(L_j[l+1].f_j - L_j[l-1].f_j)}{L_j[k] - L_j[k-1]}$
  3. Sort all individuals in "front i" according to their crowding distance *cd descendingly*

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*
- Individual A is better than individual B if it is in a lower front or if they are in the same front and A has a *higher cd*

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*
- Individual A is better than individual B if it is in a lower front or if they are in the same front and A has a *higher cd*
- Truncation selection is performed on this sorted population

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*
- Individual A is better than individual B if it is in a lower front or if they are in the same front and A has a *higher cd*
- Truncation selection is performed on this sorted population
- Elitism: Parents and children compete with each other

NSGA-II

- NSGA-II [36, 37, 57] is one of the most well-known multi-objective EAs
- Divide population into Pareto fronts according the domination criterion
- To obtain a good spread of solutions along the Pareto front, the diversity in the population must be preserved
- Within each "front i", the solutions are sorted according to their *crowding distance cd*
- Individual A is better than individual B if it is in a lower front or if they are in the same front and A has a *higher cd*
- Truncation selection is performed on this sorted population
- Elitism: Parents and children compete with each other
- Read [36, 37, 57] for more details.

Many-Objective Optimization Optimization

- Can we now solve problems with arbitrarily many optimization goals?

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]

Many-Objective Optimization Optimization

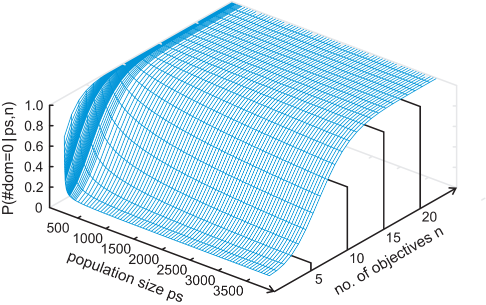
- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]

Many-Objective Optimization Optimization

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]
- When the dimension of the MOPs increases, the majority of the candidate solutions become non-dominated

Many-Objective Optimization Optimization

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]
- When the dimension of the MOPs increases, the majority of the candidate solutions become non-dominated



Fraction of non-dominated elements in ps uniform random samples of dimension n

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]
- When the dimension of the MOPs increases, the majority of the candidate solutions become non-dominated
- The increasing dimensionality of the objective space leads to three main problems [64]:

Many-Objective Optimization Optimization

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]
- When the dimension of the MOPs increases, the majority of the candidate solutions become non-dominated
- The increasing dimensionality of the objective space leads to three main problems [64]:
  1. The performance of traditional approaches based solely on Pareto comparisons deteriorates.

Many-Objective Optimization Optimization

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]
- When the dimension of the MOPs increases, the majority of the candidate solutions become non-dominated
- The increasing dimensionality of the objective space leads to three main problems [64]:
  1. The performance of traditional approaches based solely on Pareto comparisons deteriorates.
  2. The utility of the solutions cannot be understood by the human operator anymore.

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]
- When the dimension of the MOPs increases, the majority of the candidate solutions become non-dominated
- The increasing dimensionality of the objective space leads to three main problems [64]:
  1. The performance of traditional approaches based solely on Pareto comparisons deteriorates.
  2. The utility of the solutions cannot be understood by the human operator anymore.
  3. The number of possible Pareto-optimal solutions may increase exponentially.

Many-Objective Optimization Optimization

- Can we now solve problems with arbitrarily many optimization goals?
- Pareto-(ranking) based optimization performs very bad if the number n of objective functions is too high! [47, 60–62]
- Most studies consider mainly bi-objective problems [47, 63]
- When the dimension of the MOPs increases, the majority of the candidate solutions become non-dominated
- The increasing dimensionality of the objective space leads to three main problems [64]:
  1. The performance of traditional approaches based solely on Pareto comparisons deteriorates.
  2. The utility of the solutions cannot be understood by the human operator anymore.
  3. The number of possible Pareto-optimal solutions may increase exponentially.
- Therefore: Do NOT use too many objective functions!

Summary

- Many problems are multi-objective by nature, for example Genetic Programming: minimize Symbolic Regression expression size and error

Summary

- Many problems are multi-objective by nature
- If objectives conflict, a special treatment is necessary

Summary

- Many problems are multi-objective by nature
- If objectives conflict, a special treatment is necessary
- Pareto ranking is a good idea: incorporated into MOEAs

Summary

- Many problems are multi-objective by nature
- If objectives conflict, a special treatment is necessary
- Pareto ranking is a good idea: incorporated into MOEAs

Summary

- Many problems are multi-objective by nature
- If objectives conflict, a special treatment is necessary
- Pareto ranking is a good idea: incorporated into MOEAs
- Many treatments (e.g., Pareto) can be reduced to binary comparisons

谢谢

Thank you

Thomas Weise [汤卫思]
tweise@hfuu.edu.cn
http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui,
China

Caspar David Friedrich, “Der Wanderer über dem Nebelmeer”, 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog

Bibliography

1. Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley Interscience Series in Systems and Optimization. New York, NY, USA: John Wiley & Sons Ltd., May 2001. ISBN 047187339X and 978-0471873396. URL http://books.google.de/books?id=ll_Blt03u5IC.
2. Carlos Artemio Coello Coello. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In Peter John Angeline, Zbigniew Michalewicz, Marc Schoenauer, Xin Yao, and Ali M. S. Zalzala, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'99)*, volume 1, pages 3–13, Washington, DC, USA: Mayflower Hotel, July 6–9, 1999. Piscataway, NJ, USA: IEEE Computer Society. doi: 10.1109/CEC.1999.781901. URL http://www-course.cs.york.ac.uk/evo/SupportingDocs/coellocoello99updated.pdf.
3. Carlos M. Fonseca and Peter J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms – part i: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 28(1):26–37, January 1998. doi: 10.1109/3468.650319. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.2473.
4. Robin Charles Purshouse. *On the Evolutionary Optimisation of Many Objectives*. PhD thesis, Sheffield, UK: University of Sheffield, Department of Automatic Control and Systems Engineering, September 2003. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.8816.
5. Robin Charles Purshouse and Peter J. Fleming. Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)*, volume 2632/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 16–30, Faro, Portugal: University of the Algarve, April 8–11, 2003. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-36970-8_2. URL http://www.lania.mx/~ccoello/EMOO/purshouse03.pdf.gz.
6. Matthias Ehrgott. *Multicriteria Optimization*, volume 491 of *Lecture Notes in Economics and Mathematical Systems*. Basel, Switzerland: Birkhäuser Verlag, 2nd edition, 2005. ISBN 3-540-21398-8 and 978-3-540-21398-7. URL http://books.google.de/books?id=yrZw9srrHroC.
7. Johan Anderson. A survey of multiobjective optimization in engineering design. Technical Report LiTH-IKP-R-1097, Linköping, Sweden: Linköping University, Department of Mechanical Engineering, 2000. URL http://www.lania.mx/~ccoello/andersson00.pdf.gz.
8. Mark J. Rentmeesters, Wei K. Tsai, and Kwei-Jay Lin. A theory of lexicographic multi-criteria optimization. In *Second IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'96)*, pages 76–96, Montréal, QC, Canada, October 21–25, 1996. Piscataway, NJ, USA: IEEE Computer Society. doi: 10.1109/ICECCS.1996.558386.

Bibliography II

9. A. Volgenant. Solving some lexicographic multi-objective combinatorial problems. *European Journal of Operational Research (EJOR)*, 139(3):578–584, June 16, 2002. doi: 10.1016/S0377-2217(01)00214-4.
10. L. Darrell Whitley. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In James David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, pages 116–121, Fairfax, VA, USA: George Mason University (GMU), June 4–7, 1989. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. URL `http://citeseer.ist.psu.edu/531140.html`.
11. Indraneel Das and John E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, August 1997. doi: 10.1007/BF01197559.
12. Yann Collette and Patrick Siarry. *Multiobjective Optimization: Principles and Case Studies*, volume 2 of *Decision Engineering*. Paris, France: Groupe Eyrolles, 2002. ISBN 3540401822 and 978-3-540-40182-7. URL `http://books.google.de/books?id=XNYF4hltoF0C`.
13. Ramprasad S. Krishnamachari and Panos Y. Papalambros. Hierarchical decomposition synthesis in optimal systems design. *Journal of Mechanical Design*, 119(4):448–457, December 1997. doi: 10.1115/1.2826388. URL `http://ode.engin.umich.edu/publications/PapalambrosPapers/1996/102.pdf`.
14. Ramprasad S. Krishnamachari and Panos Y. Papalambros. Hierarchical decomposition synthesis in optimal systems design. Technical Report 95-16, Ann Arbor, MI, USA: University of Michigan, Department of Mechanical Engineering and Applied Mechanics, Design Laboratory, 1995. URL `http://hdl.handle.net/2027.42/6077`.
15. David T. Fullwood. Percolation in two-dimensional grain boundary structures, and polycrystal property closures. Master's thesis, Provo, UT, USA: Brigham Young University, December 2005. URL `http://contentdm.lib.byu.edu/ETD/image/etd1045.pdf`.
16. Juhani Koski. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in Applied Numerical Methods (CANM)*, 1(6):333–337, November 1985. doi: 10.1002/cnm.1630010613.
17. Achille Messac and Christopher A. Mattson. Generating well-distributed sets of pareto points for engineering design using physical programming. *Optimization and Engineering*, 3(4):431–450, December 2002. doi: 10.1023/A:1021179727569. URL `http://www.rpi.edu/~messac/Publications/messac_2002_pareto_pp_opte.pdf`.
18. Francis Ysidro Edgeworth. *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. Whitefish, MT, USA: Kessinger Publishing and London, UK: C. Kegan Paul & Co, 1881. ISBN 0548216657, 1163230006, 978-0548216651, and 978-1163230008. URL `http://socserv.mcmaster.ca/econ/ugcm/3ll3/edgeworth/mathpsychics.pdf`.

19. Vilfredo Federico Pareto. *Cours d'Économie Politique*. Lausanne/Paris, France: F. Rouge, 1896. URL http://ann.sagepub.com/cgi/reprint/9/3/128.pdf. 2 volumes.
20. Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995. doi: 10.1162/evco.1995.3.1.1. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.7779.
21. Vira Chankong and Yacov Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. Amsterdam, The Netherlands: North-Holland Scientific Publishers Ltd., Amsterdam, The Netherlands: Elsevier Science Publishers B.V., and Mineola, NY, USA: Dover Publications, January 1983. ISBN 0-444-00710-5, 978-0-444-00710-0, and 978-0-486-46289-9. URL http://books.google.de/books?id=fQ5VPQAACAAJ.
22. Ralph E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. Malabar, FL, USA: R. E. Krieger Publishing Company, reprint edition, August 1989. ISBN 0894643932 and 978-0894643934. URL http://books.google.de/books?id=r5P0AQAACAAJ.
23. Yacov Y. Haimes and Ralph E. Steuer, editors. *Proceedings of the 14th International Conference on Multiple Criteria Decision Making: Research and Practice in Multi Criteria Decision Making (MCDM'98)*, volume 487 of *Lecture Notes in Economics and Mathematical Systems*, Charlottesville, VA, USA: University of Virginia, June 12–18, 1998. Berlin/Heidelberg: Springer-Verlag. ISBN 3-540-67266-1 and 978-3-540-67266-1. URL http://books.google.de/books?id=skkEdJir95QC.
24. E. A. Galperin. Pareto analysis vis-à-vis balance space approach in multiobjective global optimization. *Journal of Optimization Theory and Applications*, 93(3):533–545, June 1997. doi: 10.1023/A:1022639028824.
25. Aharon Ben-Tal. Characterization of pareto and lexicographic optimal solutions. In Joel N. Morse, editor, *Proceedings of the 4th International Conference on Multiple Criteria Decision Making: Organizations, Multiple Agents With Multiple Criteria (MCDM'80)*, volume 190 of *Lecture Notes in Economics and Mathematical Systems*, pages 1–11, Newark, DE, USA: University of Delaware, August 10–15, 1980. Berlin/Heidelberg: Springer-Verlag.
26. Altannar Chinchuluun, Panos M. Pardalos, Athanasios Migdalas, and Leonidas S. Pitsoulis, editors. *Pareto Optimality, Game Theory and Equilibria*, volume 17 of *Springer Optimization and Its Applications*. New York, NY, USA: Springer New York, 2008. ISBN 0387772464, 978-0-387-77246-2, and 978-0-387-77247-9. doi: 10.1007/978-0-387-77247-9. URL http://books.google.de/books?id=kNqHxU3Tc2YC.
27. Wikipedia – the free encyclopedia, 2009. URL http://en.wikipedia.org/.
28. Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. Cambridge, MA, USA: MIT Press, July 1994. ISBN 0-2626-5040-1 and 978-0-262-65040-3. URL http://books.google.de/books?id=5ntdaYX4LPkC.

Bibliography IV

29. Drew Fudenberg and Jean Tirole. *Game Theory*. Cambridge, MA, USA: MIT Press, August 1991. ISBN 0-2620-6141-4 and 978-0-262-06141-4. URL http://books.google.de/books?id=pFPHKwXro3QC.
30. Thomas Weise, Michael Zapf, Raymond Chiong, and Antonio Jesús Nebro Urbaneja. Why is optimization difficult? In Raymond Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, volume 193/2009 of *Studies in Computational Intelligence*, chapter 1, pages 1–50. Berlin/Heidelberg: Springer-Verlag, April 30, 2009. doi: 10.1007/978-3-642-00267-0_1.
31. Carlos Artemio Coello Coello. Evolutionary multiobjective optimization. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5):444–447, September–October 2011. doi: 10.1002/widm.43.
32. David A. van Veldhuizen and Laurence D. Merkle. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Wright-Patterson Air Force Base, OH, USA: Air University, Air Force Institute of Technology, June 1999. URL http://handle.dtic.mil/100.2/ADA364478.
33. Carlos Artemio Coello Coello. A short tutorial on evolutionary multiobjective optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos Artemio Coello Coello, and David Wolfe Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01)*, volume 1993/2001 of *Lecture Notes in Computer Science (LNCS)*, pages 21–40, Zürich, Switzerland: Eidgenössische Technische Hochschule (ETH) Zürich, March 7–9, 2001. Berlin, Germany: Springer-Verlag GmbH. URL http://www.cs.cinvestav.mx/~emooworkgroup/tutorial-slides-coello.pdf.
34. Carlos Artemio Coello Coello. An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)*, 32(2):109–143, June 2000. doi: 10.1145/358923.358929. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.8622.
35. David A. van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, Summer 2000. doi: 10.1162/106365600568158. URL http://mitpress.mit.edu/journals/EVCO/VanVeldhuizen.pdf.
36. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. KanGAL Report 200001, Kanpur, Uttar Pradesh, India: Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur (IIT), 2000. URL http://www.jeo.org/emo/deb00.ps.gz.

37. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julián Merelo-Guervós, and Hans-Paul Schwefel, editors, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN VI)*, volume 1917/2000 of *Lecture Notes in Computer Science (LNCS)*, pages 849–858, Paris, France, September 18–20, 2000. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-45356-3_83. URL https://eprints.kfupm.edu.sa/17643/1/17643.pdf.
38. Ajith Abraham, Lakhmi C. Jain, and Robert Goldberg, editors. *Evolutionary Multiobjective Optimization – Theoretical Advances and Applications*. Advanced Information and Knowledge Processing. Berlin, Germany: Springer-Verlag GmbH. ISBN 1852337877 and 978-1-85233-787-2. URL http://books.google.de/books?id=Ei7q1YSjiSAC.
39. Carlos Artemio Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems – An International Journal (KAIS)*, 1(3), August 1999. URL http://www.lania.mx/~ccoello/EMOO/informationfinal.ps.gz.
40. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, Summer 2000. doi: 10.1162/106365600568202. URL http://sci2s.ugr.es/docencia/cursoMieres/EC-2000-Comparison.pdf.
41. David Edward Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0-201-15767-5 and 978-0-201-15767-3. URL http://books.google.de/books?id=2IIJAAAACAAJ.
42. Tapabrata Ray, Kang Tai, and Kin Chye Seow. Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*, 33(4):399–424, 2001. doi: 10.1080/03052150108940926.
43. Ruhul Amin Sarker, Hussein Aly Abbass Amein, and Samin Karim. An evolutionary algorithm for constrained multiobjective optimization problems. In Nikola Kasabov and Peter Alexander Whigham, editors, *Proceedings of the 5th Australia-Japan Joint Workshop on Intelligent & Evolutionary Systems – "From Population Genetics to Evolving Intelligent Systems" (AJWIS'01)*, pages 113–122, Dunedin, New Zealand: University of Otago, November 19–21, 2001. URL http://www.lania.mx/~ccoello/EMOO/sarker01a.pdf.gz.
44. Lothar Thiele, Kaisa Miettinen, Pekka J. Korhonen, and Julian Molina. A preference-based interactive evolutionary algorithm for multiobjective optimization. HSE Working Paper W-412, Helsinki, Finland: Helsinki School of Economics (HSE, Helsingin kauppakorkeakoulu), January 2007. URL http://hsepubl.lib.hse.fi/pdf/wp/w412.pdf.

45. Yonas Gebre Woldesenbet, Gary G. Yen, and Biruk G. Tessema. Constraint handling in multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 13(3):514–525, June 2009. doi: 10.1109/TEVC.2008.2009032.
46. Vineet Khare, Xin Yao, and Kalyanmoy Deb. Performance scaling of multi-objective evolutionary algorithms. Technical Report 2002009, Kanpur, Uttar Pradesh, India: Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur (IIT), October 2002. URL http://citeseer.ist.psu.edu/old/597663.html.
47. Vineet Khare, Xin Yao, and Kalyanmoy Deb. Performance scaling of multi-objective evolutionary algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)*, volume 2632/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 367–390, Faro, Portugal: University of the Algarve, April 8–11, 2003. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-36970-8_27.
48. Sanaz Mostaghim. *Multi-objective Evolutionary Algorithms: Data structures, Convergence and, Diversity*. PhD thesis, Paderborn, Germany: Universität Paderborn, Fakultät für Elektrotechnik, Informatik und Mathematik, February 2005. URL http://books.google.de/books?id=HoVLAAAACAAJ.
49. Robin Charles Purshouse and Peter J. Fleming. The multi-objective genetic algorithm applied to benchmark problems – an analysis. Research Report 796, Sheffield, UK: University of Sheffield, Department of Automatic Control and Systems Engineering, August 2001. URL http://www.lania.mx/~ccoello/EMOO/purshouse01.pdf.gz.
50. Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Ralph E. Steuer, editors. *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Wadern, Germany: Schloss Dagstuhl, November 7–12, 2004. Schloss Dagstuhl, Wadern, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI). URL http://drops.dagstuhl.de/portals/index.php?semnr=04461. Published in 2005.
51. Dhish Kumar Saxena and Kalyanmoy Deb. Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective. In Zbigniew Michalewicz and Robert G. Reynolds, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'08), Computational Intelligence: Research Frontiers – IEEE World Congress on Computational Intelligence – Plenary/Invited Lectures (WCCI)*, volume 5050/2008 of *Lecture Notes in Computer Science (LNCS)*, pages 3204–3211, Hong Kong (Xianggang), China: Hong Kong Convention and Exhibition Centre, June 1–6, 2008. Piscataway, NJ, USA: IEEE Computer Society. doi: 10.1109/CEC.2008.4631232.

Bibliography VII

52. André Sülflow, Nicole Drechsler, and Rolf Drechsler. Robust multi-objective optimization in high dimensional spaces. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO'07)*, volume 4403/2007 of *Lecture Notes in Computer Science (LNCS)*, pages 715–726, Matsushima, Sendai, Japan, March 5–8, 2007. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/978-3-540-70928-2_54. URL http://www.informatik.uni-bremen.de/agra/doc/konf/emo2007_RobustOptimization.pdf.
53. Thomas Weise, Stefan Niemczyk, Hendrik Skubch, Roland Reichle, and Kurt Geihs. A tunable model for multi-objective, epistatic, rugged, and neutral fitness landscapes. In Maarten Keijzer, Giuliano Antoniol, Clare Bates Congdon, Kalyanmoy Deb, Benjamin Doerr, Nikolaus Hansen, John H. Holmes, Gregory S. Hornby, Daniel Howard, James Kennedy, Sanjeev P. Kumar, Fernando G. Lobo, Julian Francis Miller, Jason H. Moore, Frank Neumann, Martin Pelikan, Jordan B. Pollack, Kumara Sastry, Kenneth Owen Stanley, Adrian Stoica, El-Ghazali Talbi, and Ingo Wegener, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08)*, pages 795–802, Atlanta, GA, USA: Renaissance Atlanta Hotel Downtown, July 12–16, 2008. New York, NY, USA: ACM Press. doi: 10.1145/1389095.1389252.
54. Haiming Lu. *State-of-the-Art Multiobjective Evolutionary Algorithms – Pareto Ranking, Density Estimation and Dynamic Population*. PhD thesis, Stillwater, OK, USA: Oklahoma State University, Faculty of the Graduate College, August 2002. URL http://www.lania.mx/~ccoello/EMOO/thesis_lu.pdf.gz.
55. Garrison W. Greenwood, Xiaobo Sharon Hu, and Joseph G. D'Ambrosio. Fitness functions for multiple objective optimization problems: Combining preferences with pareto rankings. In Richard K. Belew and Michael D. Vose, editors, *Proceedings of the 4th Workshop on Foundations of Genetic Algorithms (FOGA'96)*, pages 437–455, San Diego, CA, USA: University of San Diego, August 5, 1996. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
56. Mikkel T. Jensen. Reducing the run-time complexity of multiobjective eas: The nsga-ii and other algorithms. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 7(5):503–515, October 2003. doi: 10.1109/TEVC.2003.817234.
57. Kalyanmoy Deb, Amrit Pratab, Samir Agrawal, and T Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 6(2):182–197, April 2002. doi: 10.1109/4235.996017. URL http://dynamics.org/~altenber/UH_ICS/EC_REFS/MULTI_OBJ/DebPratapAgarwalMeyarivan.pdf.

58. Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. TIK-Report 101, Zürich, Switzerland: Eidgenössische Technische Hochschule (ETH) Zürich, Department of Electrical Engineering, Computer Engineering and Networks Laboratory (TIK), May 2001. URL `http://www.tik.ee.ethz.ch/sop/publicationListFiles/zlt2001a.pdf`. Errata added 2001-09-27.
59. David Wolfe Corne, Joshua D. Knowles, and Martin J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julián Merelo-Guervós, and Hans-Paul Schwefel, editors, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN VI)*, volume 1917/2000 of *Lecture Notes in Computer Science (LNCS)*, pages 839–848, Paris, France, September 18–20, 2000. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-45356-3_82. URL `http://www.lania.mx/~ccoello/corne00.ps.gz`.
60. Vineet Khare. Performance scaling of multi-objective evolutionary algorithms. Master's thesis, Birmingham, UK: University of Birmingham, School of Computer Science, September 21, 2002. URL `http://www.lania.mx/~ccoello/EMOO/thesis_khare.pdf.gz`.
61. Andrzej Jaszkiewicz. On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study. *European Journal of Operational Research (EJOR)*, 158(2):418–433, October 16, 2004. doi: 10.1016/j.ejor.2003.06.015.
62. Hisao Ishibuchi, Yusuke Nojima, and Tsutomu Doi. Comparison between single-objective and multi-objective genetic algorithms: Performance comparison and performance measures. In Gary G. Yen, Simon M. Lucas, Gary B. Fogel, Graham Kendall, Ralf Salomon, Byoung-Tak Zhang, Carlos Artemio Coello Coello, and Thomas Philip Runarsson, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'06), 2006 IEEE World Congress on Computation Intelligence (WCCI'06)*, pages 3959–3966, Vancouver, BC, Canada: Sheraton Vancouver Wall Centre Hotel, July 16–21, 2006. Piscataway, NJ, USA: IEEE Computer Society, Piscataway, NJ, USA: IEEE Computer Society. doi: 10.1109/CEC.2006.1688438.
63. Tomoyuki Hiroyasu, Hiroyuki Ishida, Mitsunori Miki, and Hisatake Yokouchi. Difficulties of evolutionary many-objective optimization. Intelligent Systems Design Laboratory Research Reports (ISDL Reports) 20081006004, Kyoto, Japan: Doshisha University, Department of Knowledge Engineering and Computer Sciences, Intelligent Systems Design Laboratory, October 13, 2008. URL `http://mikilab.doshisha.ac.jp/dia/research/report/2008/1006/004/report20081006004.html`.

Bibliography IX

64. Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In Zbigniew Michalewicz and Robert G. Reynolds, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'08), Computational Intelligence: Research Frontiers – IEEE World Congress on Computational Intelligence – Plenary/Invited Lectures (WCCI)*, volume 5050/2008 of *Lecture Notes in Computer Science (LNCS)*, pages 2424–2431, Hong Kong (Xianggang), China: Hong Kong Convention and Exhibition Centre, June 1–6, 2008. Piscataway, NJ, USA: IEEE Computer Society. doi: 10.1109/CEC.2008.4631121. URL http://www.ie.osakafu-u.ac.jp/~hisaoi/ci_lab_e/research/pdf_file/multiobjective/CEC2008_Many_Objective_Final.pdf.