technische universität
dortmund

Computational Intelligence

Winter Term 2020/21

Prof. Dr. Günter Rudolph

Lehrstuhl für Algorithm Engineering (LS 11)

Fakultät für Informatik

TU Dortmund

Slides prepared by
Dr. Nicola Beume
(2012)

enriched with slides by
Prof. Dr. Boris Naujoks, TH Cologne
from Winter Term 2017/18

(with permission)

The regular optimisation problem

- Minimize

$$f : \mathcal{X} \subset \boldsymbol{R}^n \longrightarrow \mathcal{Y} \subset \boldsymbol{R}$$

- Subject to
  - Equality constraints

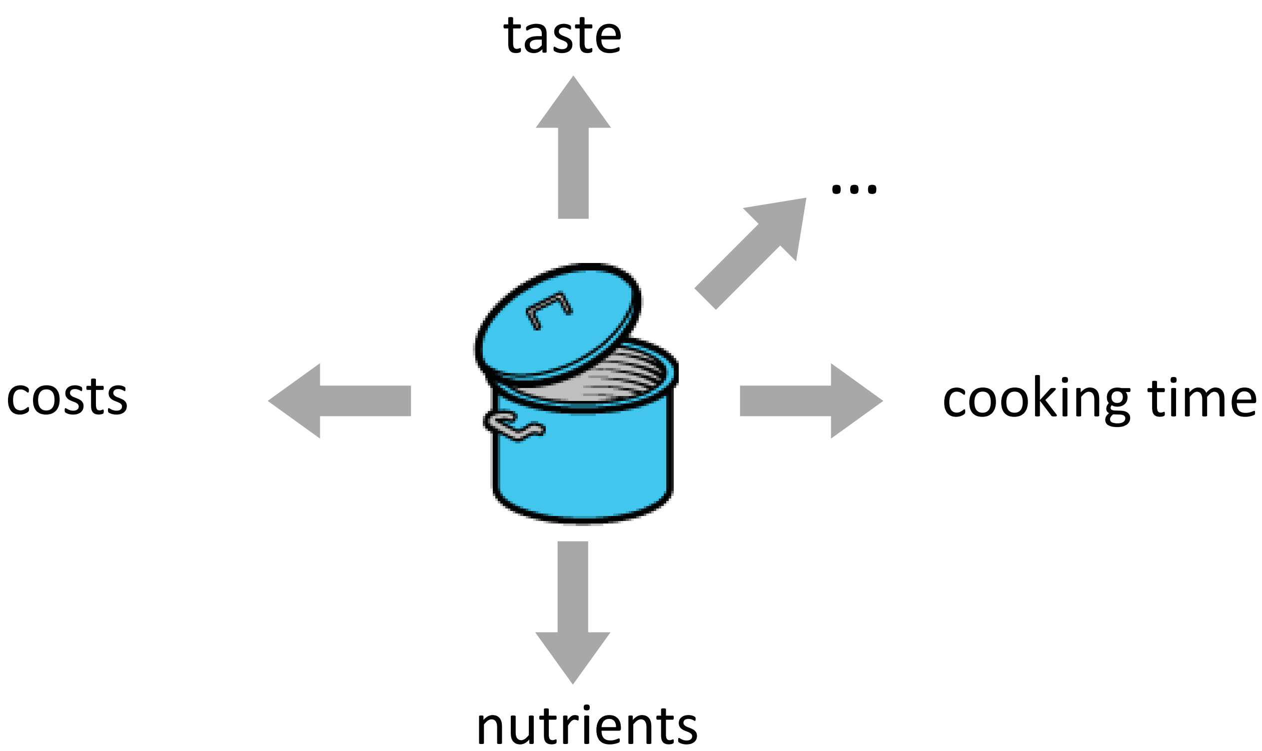
$$h(x) = 0 \qquad \forall x \in \mathcal{X}$$

  - Inequality constraints

$$g(x) \leqslant 0 \qquad \forall x \in \mathcal{X}$$

- **Definitions**
  - $x \in \mathcal{X}$ is (valid) solution
  - $\mathcal{X}$ search, parameter, or decision space
  - $\mathcal{Y}$ objective space

Multiobjective Optimization



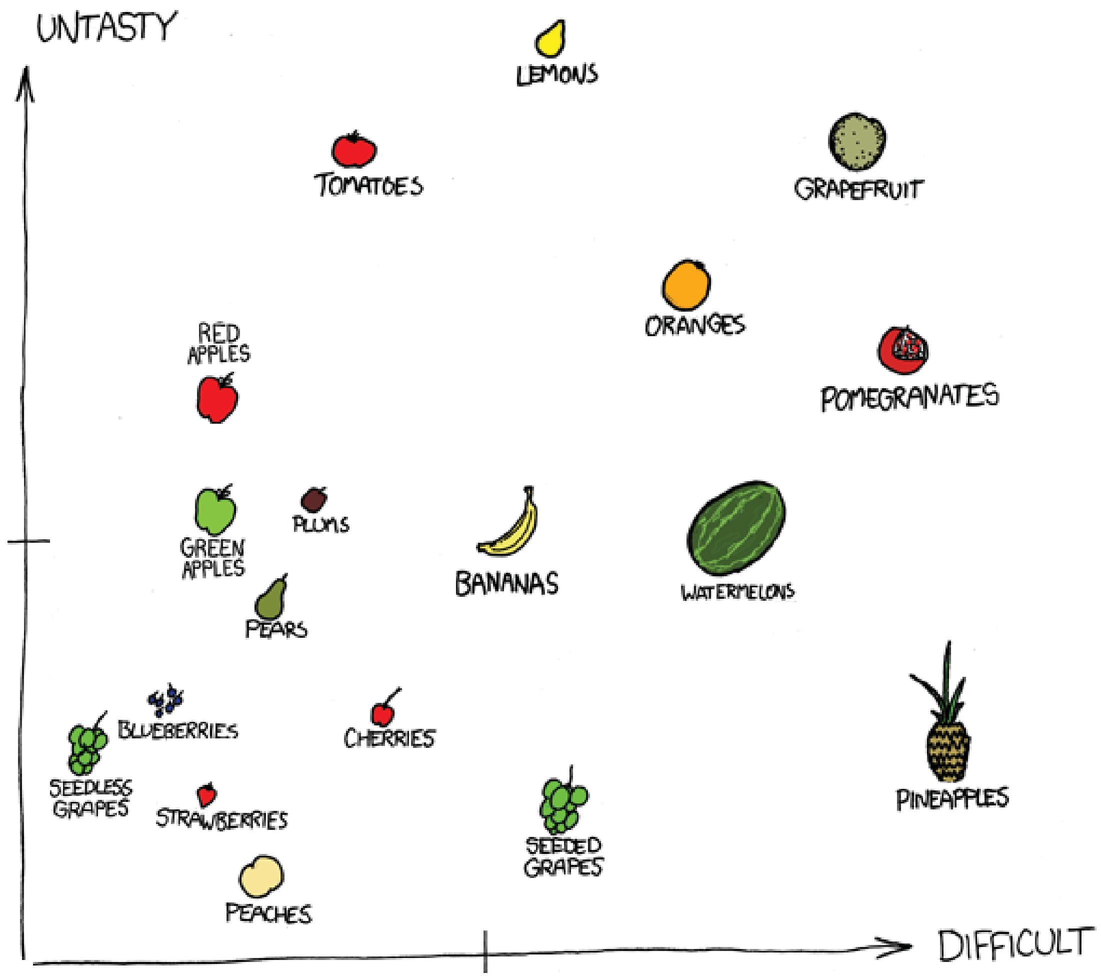
Real-world problems: various demands on problem solution

$\Rightarrow$ multiple conflictive objective functions

Laptop Selection

Name	Display	Battery	Weight	Price	CPU	RAM	Graphic	Disk	Interfaces
Dell Vostro 15 5568	15.6	12 h	2 kg	689	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, HDMI, USB
HP 14-bs007ng	14	12.5 h	1,7 kg	699	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, HDMI, USB
HP 250 2HG71ES	15.6	12 h	1,86 kg	649	I5-7	8 GB DDR4	Radeon 520	256 SSD	VGA, HDMI, USB
Lenovo ThinkPad L470	14	10 h	1,87 kg	699	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, Disp, USB
Fuijitsu Lifebook A557	15.6	8 h	2,4 kg	650	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, HDMI, USB
Levono ThinkPad E470	14	8 h	1.87 kg	699	I5-7	8 GB DDR4	GeForce 940MX	256 SSD	HDMI, USB
Levono ThinkPad E470	14	8 h	1.87 kg	849	I7-7	16 GB DDR4	GeForce 940MX	256 SSD	HDMI, USB
Lenovo ThinkPad L570	15.6	10 h	2.38 kg	849	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA; Disp, USB
Lenovo ThinkPad 13	13.3	12 h	1.44 kg	869	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	HDMI, USB
HP Power Pavilion 14-cb013ng	15.6	14.5 h	2.21 kg	1139	I7-7	16 GB DDR4	GeForce CTX 1050 Ti	256 SSD + 1T HDD	HDMI, USB
HP Power Pavilion 15-cb013ng	15.6	14.5 h	2.21 kg	1139	I7-7	16 GB DDR4	GeForce GtX 1050 Ti	256 SSD + 1T HDD	HDMI, USB
Asus X556UQ-DM885T	15.6	decent	2.3 kg	719	I5-7	8 GB DDR4	GeForce 940MX	256 SSD + 1T HDD	VGA; HDMI; USB
Acer Aspire 5 A515-51G-51 RL	15.6	9 h	2.1 kg	849	I5-7	8 GB DDR4	Geforce MX150	128 SSD + 1T HDD	HDMI; USB
HP Pavilion 14-bf007ng	14	10.25 h	1.53 kg	666	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	HDMI; USB
Acer Swift 3 (SF314-51-77W2)	14	10 h	1.65 kg	774	I7-7	8 GB DDR4	HD Graphics 620	256 SSD	HDMI; USB
Lenovo ThinkPad X1 Carbon	14	12 h	1.13 kg	879	I7-4	8 GB DDR3L	HD Graphics 5000	256 SSD	HDMI; DISP; USB
Fujitsu Lifebook A557	15.6	8 h	2.4 kg	613	I5-7	16 GB DDR4	HD Graphics 620	512 SSD	VGA; HDMI; USB
Acer TravelMate P459-G2-M-56T4	15.6	8 h	2.1 kg	694	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA; HDMI; USB
AsusZenbook UX3410UQ-GV999T	14	8.5 h	1.4 kg	999	I5-7	8 GB DDR4	GeForce 940MX	256 SSD + 1T HDD	HDMI; USB

Comparing Apples and Oranges

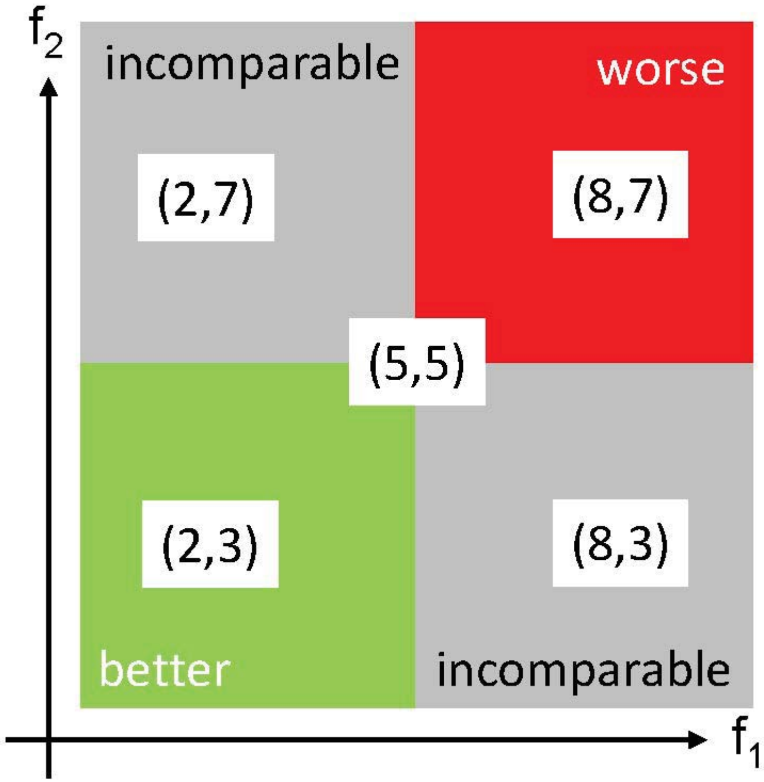


Von: `http://xkcd.com/388/`, modified

Multiobjective Optimization

Multiobjective Problem

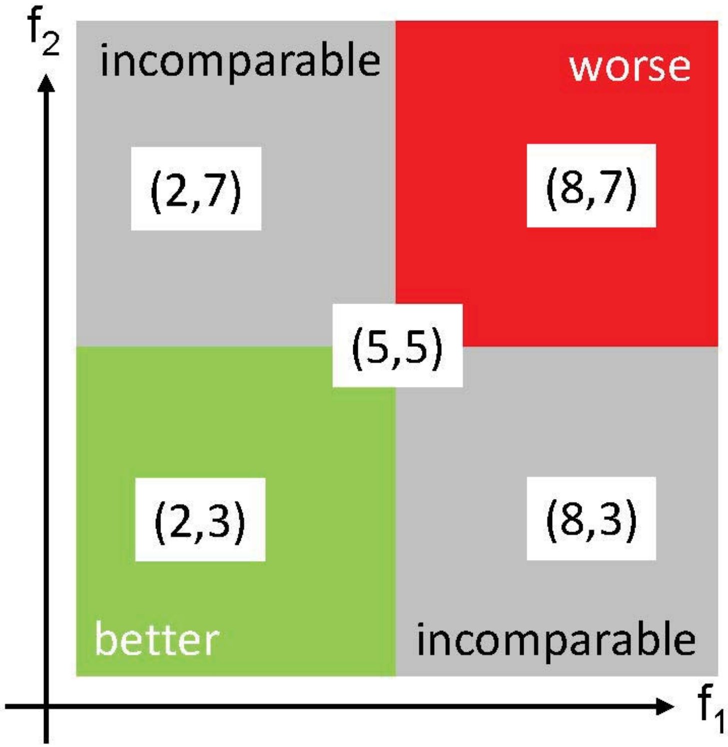
$f : S \subseteq \mathbb{R}^n \rightarrow Z \subseteq \mathbb{R}^d, \quad \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_d(\mathbf{x}))$



How to relate vectors?

Pareto Dominance

partial order among vectors in $\mathbb{R}^d$ and thus in $\mathbb{R}^n$



$$(2,3) \prec (5,5) \prec (8,7)$$
$$(2,7) \parallel (5,5) \parallel (8,3)$$

$\mathbf{a} \preceq \mathbf{b}$, $\mathbf{a}$ weakly dominates $\mathbf{b}$: $\Longleftrightarrow \forall i \in \{1,\ldots,d\} : a_i \leq b_i$

$\mathbf{a} \prec \mathbf{b}$, $\mathbf{a}$ dominates $\mathbf{b}$: $\Longleftrightarrow$ $\mathbf{a} \preceq \mathbf{b}$ and $\mathbf{a} \neq \mathbf{b}$, i.e., $\exists i \in \{1,\ldots,d\} : a_i < b_i$

$\mathbf{a} || \mathbf{b}$, $\mathbf{a}$ and $\mathbf{b}$ are incomparable: $\Longleftrightarrow$ neither $\mathbf{a} \preceq \mathbf{b}$ nor $\mathbf{b} \preceq \mathbf{a}$.

Laptop Selection

Name	Display	Battery	Weight	Price	CPU	RAM	Graphic	Disk	Interfaces
Dell Vostro 15 5568	15.6	12 h	2 kg	689	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, HDMI, USB
HP 14-bs007ng	14	12.5 h	1,7 kg	699	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, HDMI, USB
HP 250 2HG71ES	15.6	12 h	1,86 kg	649	I5-7	8 GB DDR4	Radeon 520	256 SSD	VGA, HDMI, USB
Lenovo ThinkPad L470	14	10 h	1,87 kg	699	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, Disp, USB
Fuijitsu Lifebook A557	15.6	8 h	2,4 kg	650	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA, HDMI, USB
Levono ThinkPad E470	14	8 h	1.87 kg	699	I5-7	8 GB DDR4	GeForce 940MX	256 SSD	HDMI, USB
Levono ThinkPad E470	14	8 h	1.87 kg	849	I7-7	16 GB DDR4	GeForce 940MX	256 SSD	HDMI, USB
Lenovo ThinkPad L570	15.6	10 h	2.38 kg	849	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA; Disp, USB
Lenovo ThinkPad 13	13.3	12 h	1.44 kg	869	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	HDMI, USB
HP Power Pavilion 14-cb013ng	15.6	14.5 h	2.21 kg	1139	I7-7	16 GB DDR4	GeForce CTX 1050 Ti	256 SSD + 1T HDD	HDMI, USB
HP Power Pavilion 15-cb013ng	15.6	14.5 h	2.21 kg	1139	I7-7	16 GB DDR4	GeForce GtX 1050 Ti	256 SSD + 1T HDD	HDMI, USB
Asus X556UQ-DM885T	15.6	decent	2.3 kg	719	I5-7	8 GB DDR4	GeForce 940MX	256 SSD + 1T HDD	VGA; HDMI; USB
Acer Aspire 5 A515-51G-51 RL	15.6	9 h	2.1 kg	849	I5-7	8 GB DDR4	Geforce MX150	128 SSD + 1T HDD	HDMI; USB
HP Pavilion 14-bf007ng	14	10.25 h	1.53 kg	666	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	HDMI; USB
Acer Swift 3 (SF314-51-77W2)	14	10 h	1.65 kg	774	I7-7	8 GB DDR4	HD Graphics 620	256 SSD	HDMI; USB
Lenovo ThinkPad X1 Carbon	14	12 h	1.13 kg	879	I7-4	8 GB DDR3L	HD Graphics 5000	256 SSD	HDMI; DISP; USB
Fujitsu Lifebook A557	15.6	8 h	2.4 kg	613	I5-7	16 GB DDR4	HD Graphics 620	512 SSD	VGA; HDMI; USB
Acer TravelMate P459-G2-M-56T4	15.6	8 h	2.1 kg	694	I5-7	8 GB DDR4	HD Graphics 620	256 SSD	VGA; HDMI; USB
AsusZenbook UX3410UQ-GV999T	14	8.5 h	1.4 kg	999	I5-7	8 GB DDR4	GeForce 940MX	256 SSD + 1T HDD	HDMI; USB

Aim of Optimization

Pareto front: set of optimal solution vectors in $\mathbb{R}^d$, i.e.,

PF = $\{\mathbf{x} \in Z \mid \nexists \mathbf{x}' \in Z$ with $\mathbf{x}' \prec \mathbf{x}\}$

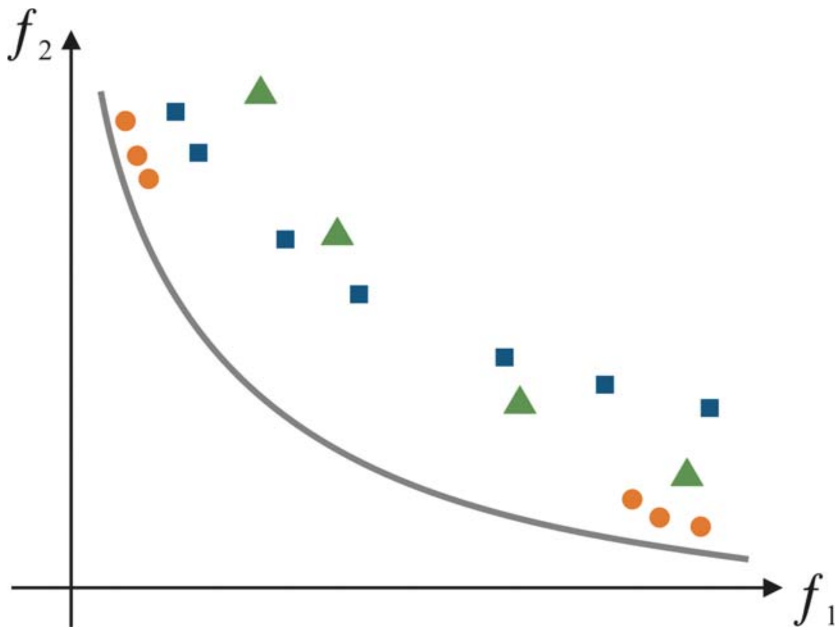
Aim of optimization: find Pareto front?

PF maybe infinitively large

PF hard to hit exactly in continuous space

$\Rightarrow$too ambitious!

Aim of optimization: approximate Pareto front!

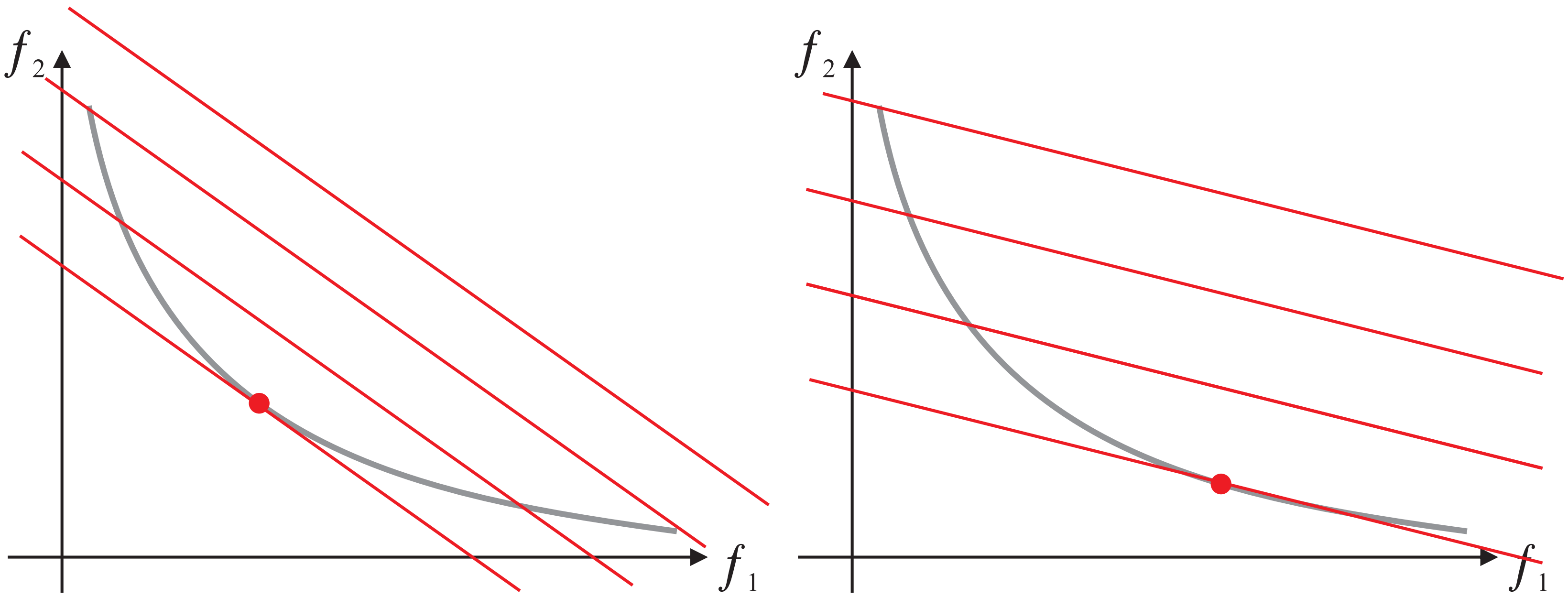


Scalarization

Isn't there an easier way?

Scalarize objectives to single-objective function:
$f : S \subseteq \mathbb{R}^n \to Z \subseteq \mathbb{R}^2 \Rightarrow \quad f_{scal} = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x})$

Result: single solution
Specify desired solution by choice of w_1, w_2



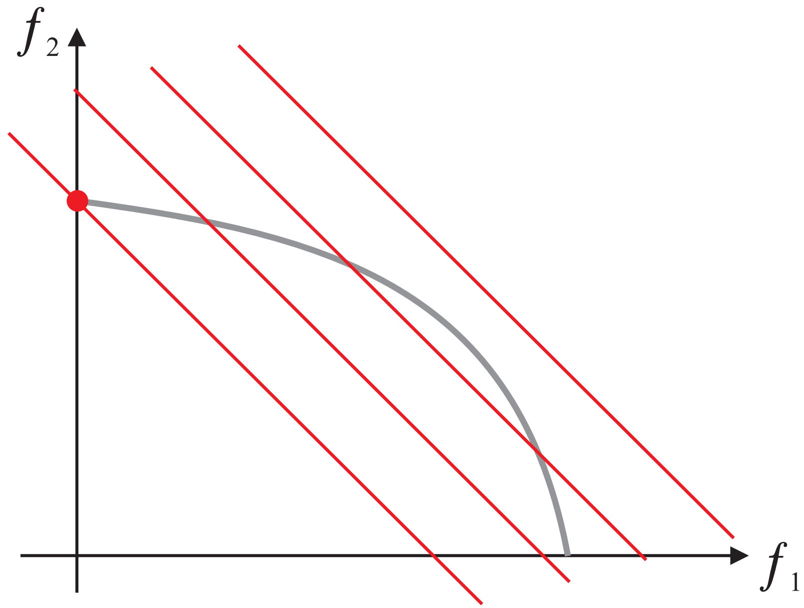
Scalarization

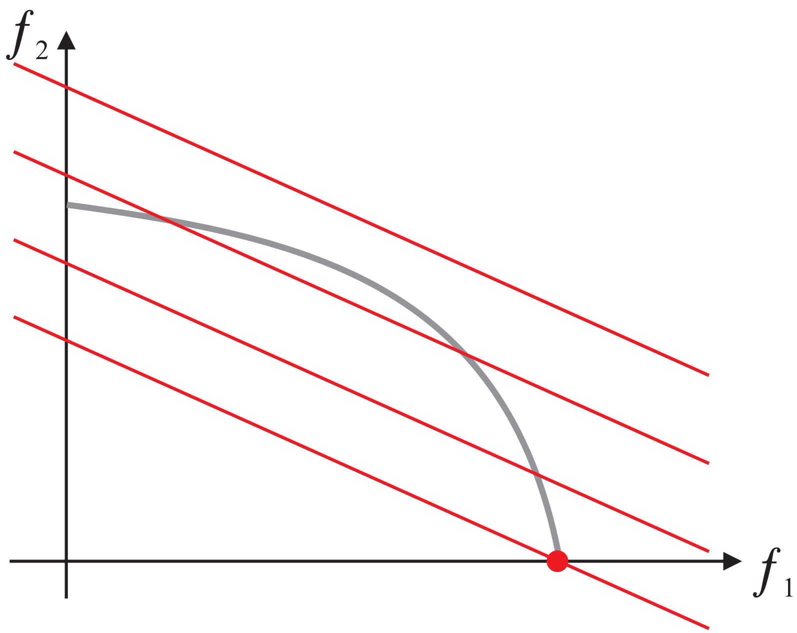
Previous example: convex Pareto front

Consider concave Pareto front

↯ only boundary solutions are optimal

⇒ scalarization by simple weighting is not a good idea





Classification

a-priori approach

first specify preferences, then optimize

more advanced scalarization techniques (e.g. Tschebyscheff)

allow to access all elements of PF

remaining difficulty:

how to express your desires through parameter values!?

a-posteriori approach

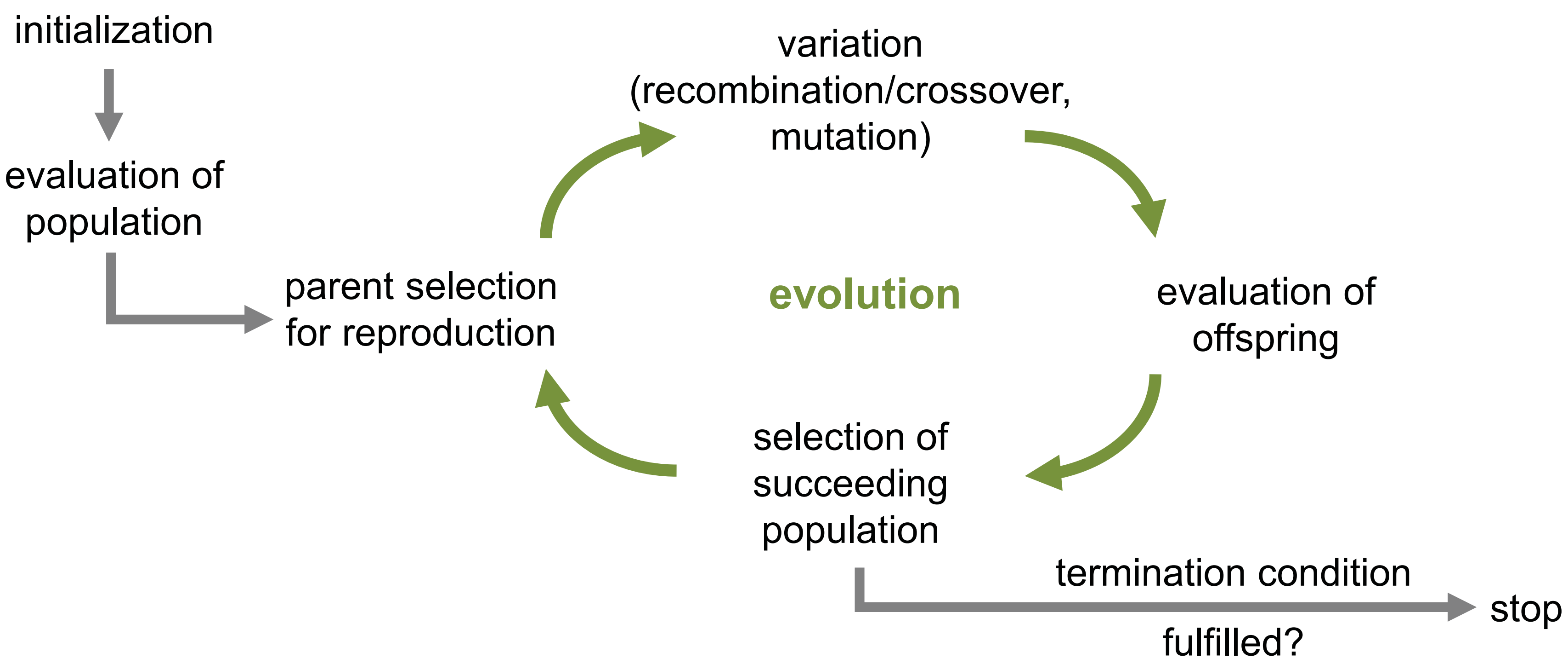
first optimize (approximate Pareto front), then choose solution

⇒back to a-posteriori approach

⇒state-of-the-art methods: evolutionary algorithms

Evolutionary Algorithms

Evolutionary Multiobjective Optimization Algorithms (EMOA)
Multiobjective Optimization Evolutionary Algorithms (MOEA)



What to change in case of multiobjective optimization?
Selection!
Remaining operators may work on search space only

Selection in EMOA

Selection requires sortable population to choose best individuals

How to sort d-dimensional objective vectors?

Primary selection criterion:
use Pareto dominance relation to sort comparable individuals

Secondary selection criterion:
apply additional measure to incomparable individuals to enforce order

Non-dominated Sorting

Example for primary selection criterion

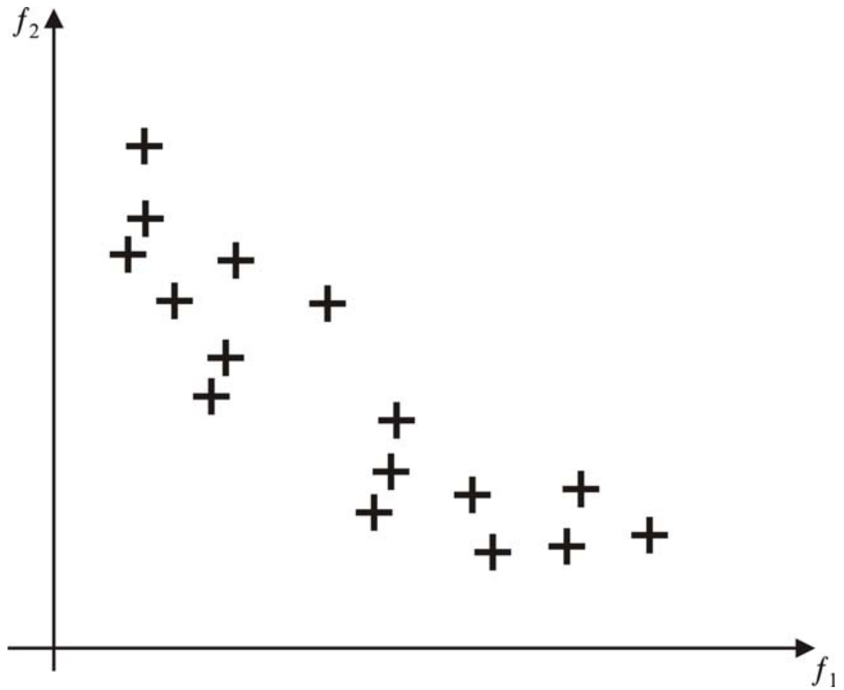
partition population into sets of mutually incomparable solutions (antichains)

non-dominated set: best elements of set

NDS(M) = $\{\mathbf{x} \in M \mid \nexists \mathbf{x}' \in M \text{ with } \mathbf{x}' \prec \mathbf{x}\}$

Simple algorithm:

iteratively remove non-dominated set until population empty



Non-dominated Sorting

Example for primary selection criterion

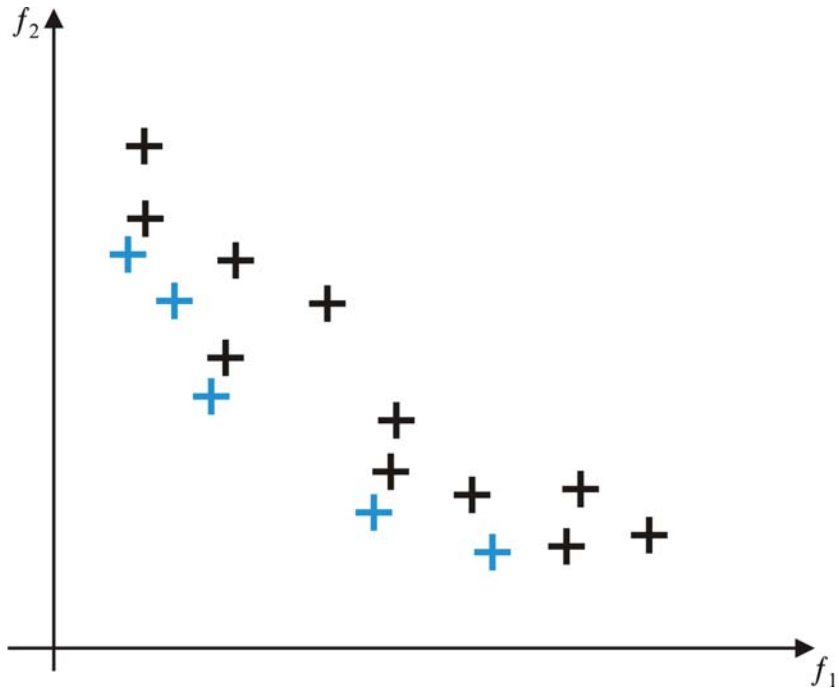
partition population into sets of mutually incomparable solutions (antichains)

non-dominated set: best elements of set

NDS(M) = $\{\mathbf{x} \in M \mid \nexists \mathbf{x}' \in M \text{ with } \mathbf{x}' \prec \mathbf{x}\}$

Simple algorithm:

iteratively remove non-dominated set until population empty



Non-dominated Sorting

Example for primary selection criterion

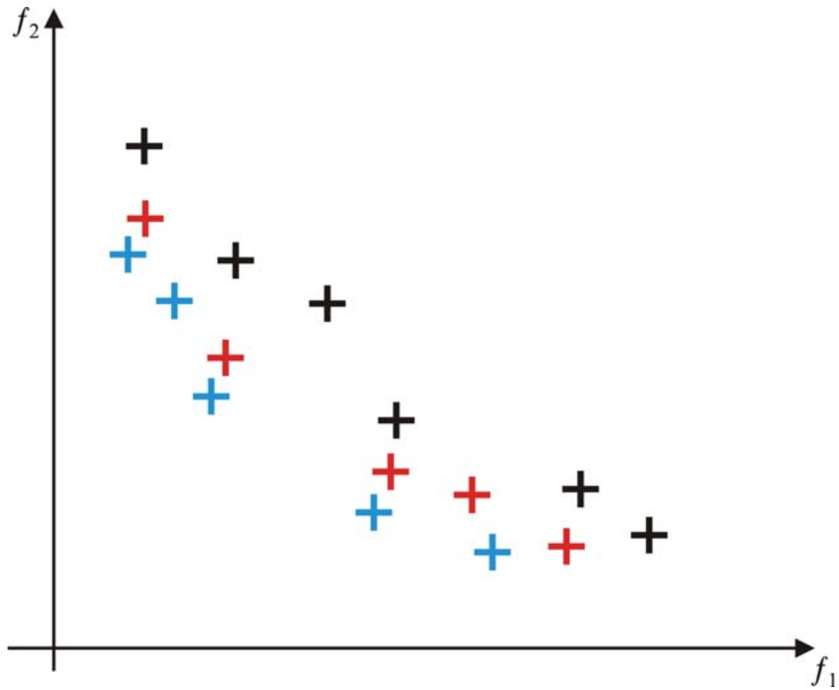
partition population into sets of mutually incomparable solutions (antichains)

non-dominated set: best elements of set

NDS(M) = $\{\mathbf{x} \in M \mid \nexists \mathbf{x}' \in M \text{ with } \mathbf{x}' \prec \mathbf{x}\}$

Simple algorithm:

iteratively remove non-dominated set until population empty



NSGA-II

Popular EMOA: Non-dominated Sorting Genetic Algorithm II

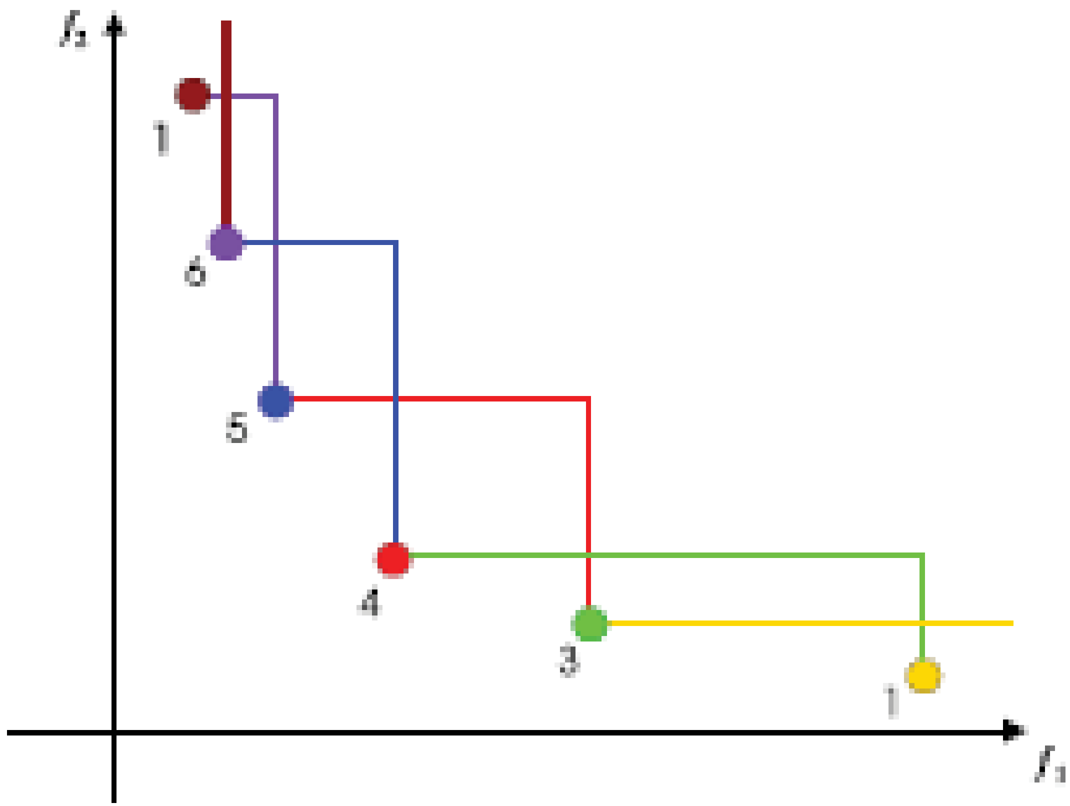
$(\mu + \mu)$-selection:

1. perform non-dominated sorting on all $\mu + \mu$ individuals
2. take best subsets as long as they can be included completely
3. if population size μ not reached but next subset does not fit in completely:
   apply secondary selection criterion *crowding distance* to that subset
4. fill up population with best ones w.r.t. the *crowding distance*

NSGA-II

Crowding distance:

- 1/2 perimeter of empty bounding box around point
- value of infinity for boundary points
- large values good



Difficulties of Selection

imagine point in the middle of the search space

$d = 2$: 1/4 better, 1/4 worse, 1/2 incomparable

$d = 3$: 1/8 better, 1/8 worse, 3/4 incomparable

general: fraction 2^{-d+1} comparable, decreases exponentially

⇒typical case: all individuals incomparable

⇒mainly secondary selection criterion in operation

Drawback of crowding distance:

rewards spreading of points, does not reward approaching the Pareto front

⇒NSGA-II diverges for large d, difficulties already for $d = 3$

Difficulties of Selection

Secondary selection criterion has to be meaningful!

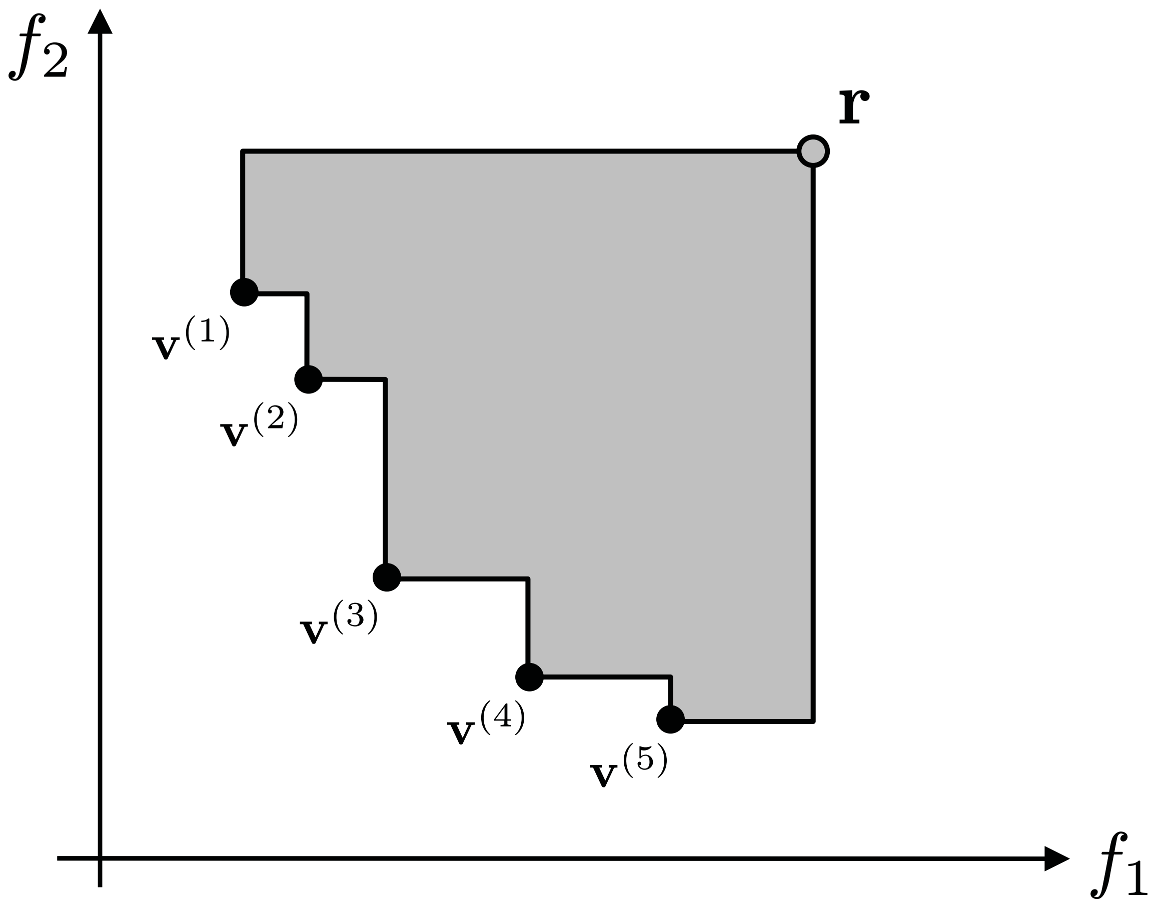
Desired: choose best subset of size μ from individuals

How to compare sets of partially incomparable points?
⇒use quality indicators for sets

One approach for selection
⇒for each point: determine contribution to quality value of set
⇒sort points according to contribution

Hypervolumen (S-metric) as Quality Measure

dominated hypervolume:
size of dominated space bounded by reference point



$$H(M, \mathbf{r}) := \text{Leb}\left(\bigcup_{i=1}^{m} [\mathbf{v}^{(i)}, \mathbf{r}]\right)$$

$$M = \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(m)}\}$$

$\mathbf{r}$ reference point

to be maximized

Example

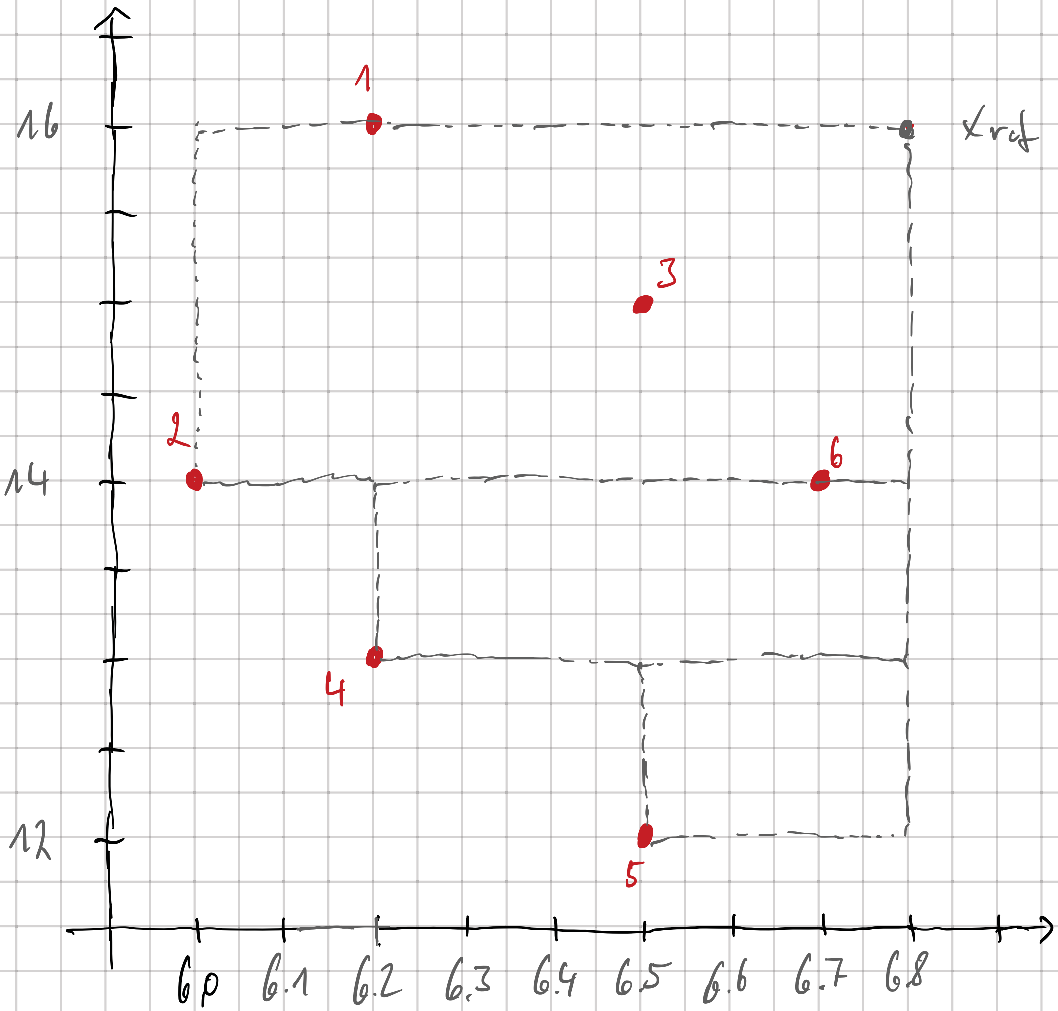
- Given the following table

Car	1	2	3	4	6	7
Consumption (l/100 km)	6.2	6.0	6.5	6.2	6.5	6.7
Price (T Euro)	16	14	15	13	12	14

- Draw the cars in objective space

- Calculate the hypervolume of the set wrt reference point $(6.8; 16)$

Example



Hypervolume: $2 \cdot 0.8 + 1 \cdot 0.6 + 1 \cdot 0.3$

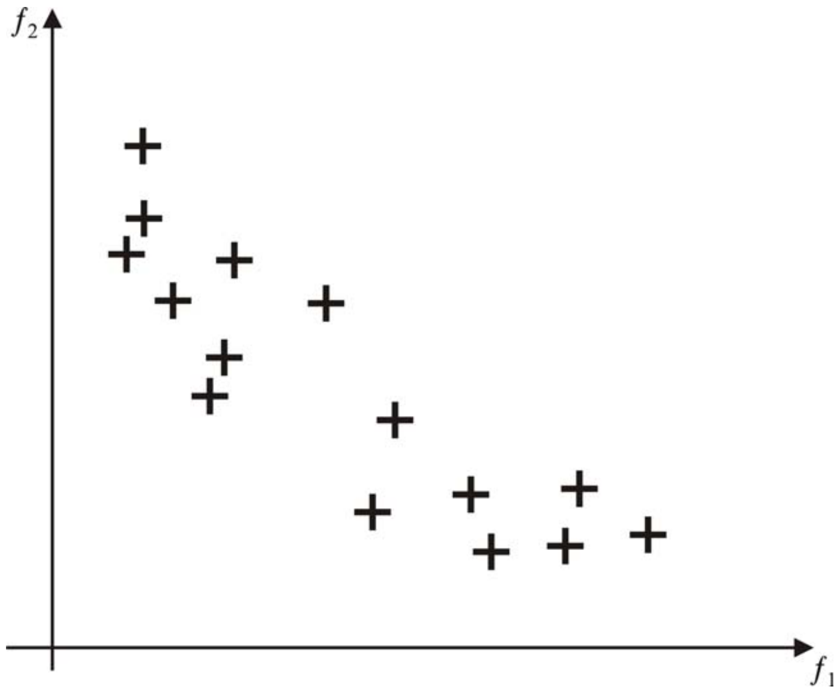
$= 1.6 + 0.6 + 0.3 = 2.5$

SMS(S-Metric Selection)-EMOA

State-of-the-art EMOA

$(\mu + 1)$-selection

1. non-dominated sorting
2. in case of incomparability: contributions to hypervolume of subset

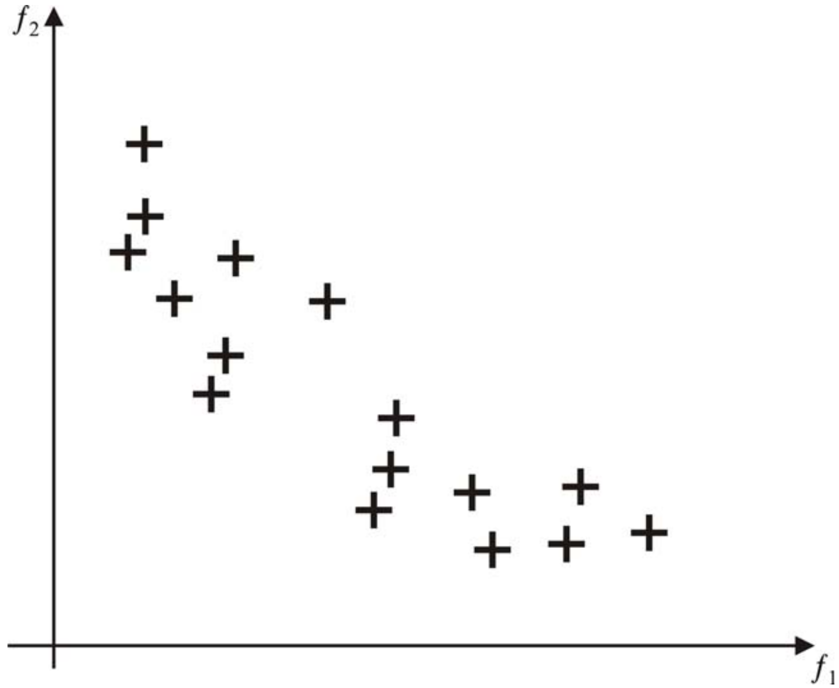


SMS(S-Metric Selection)-EMOA

State-of-the-art EMOA

$(\mu + 1)$-selection

1. non-dominated sorting
2. in case of incomparability: contributions to hypervolume of subset

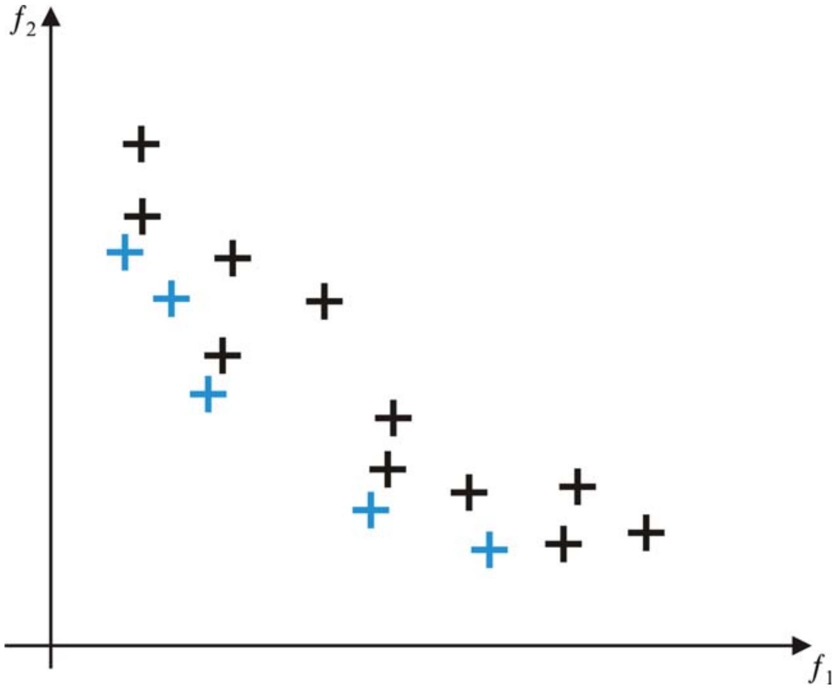


SMS(S-Metric Selection)-EMOA

State-of-the-art EMOA

$(\mu + 1)$-selection

1. non-dominated sorting
2. in case of incomparability: contributions to hypervolume of subset

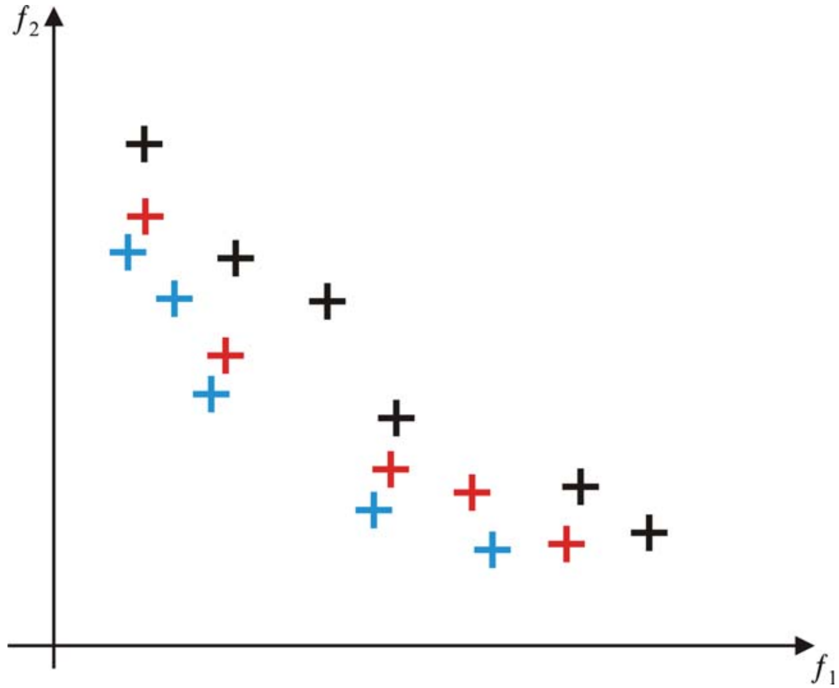


SMS(S-Metric Selection)-EMOA

State-of-the-art EMOA

$(\mu + 1)$-selection

1. non-dominated sorting
2. in case of incomparability: contributions to hypervolume of subset

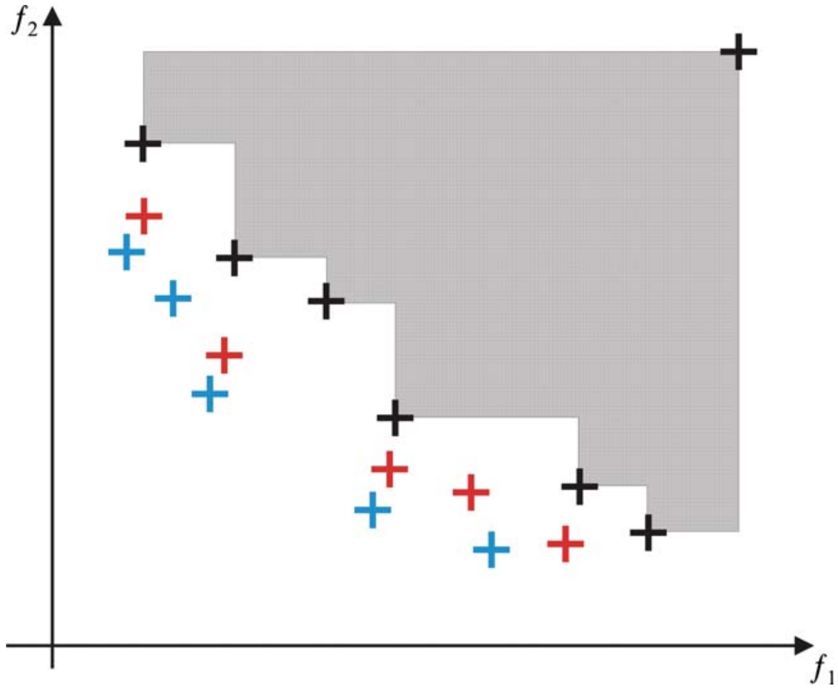


SMS(S-Metric Selection)-EMOA

State-of-the-art EMOA

$(\mu + 1)$-selection

1. non-dominated sorting
2. in case of incomparability: contributions to hypervolume of subset

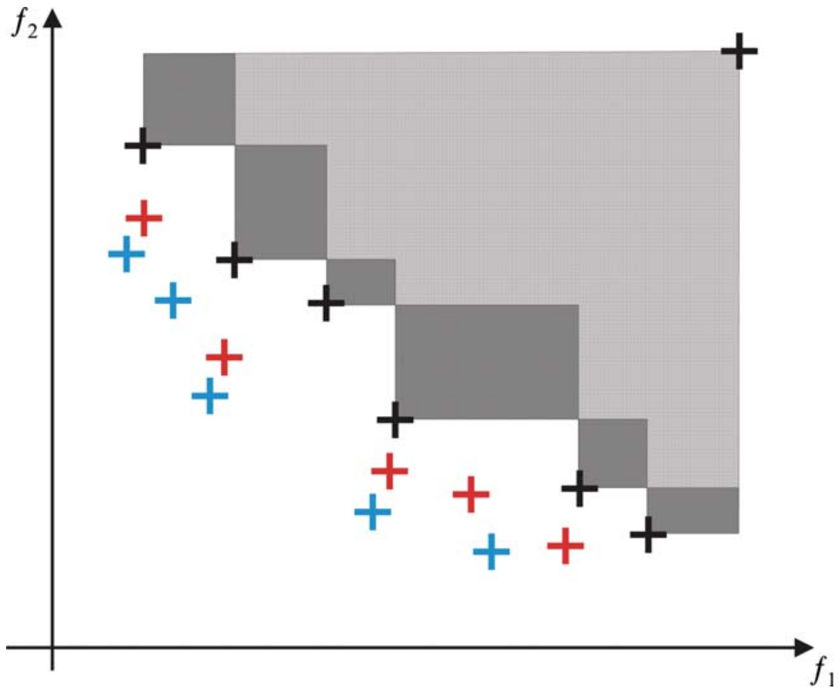


SMS(S-Metric Selection)-EMOA

State-of-the-art EMOA

$(\mu + 1)$-selection

1. non-dominated sorting
2. in case of incomparability: contributions to hypervolume of subset

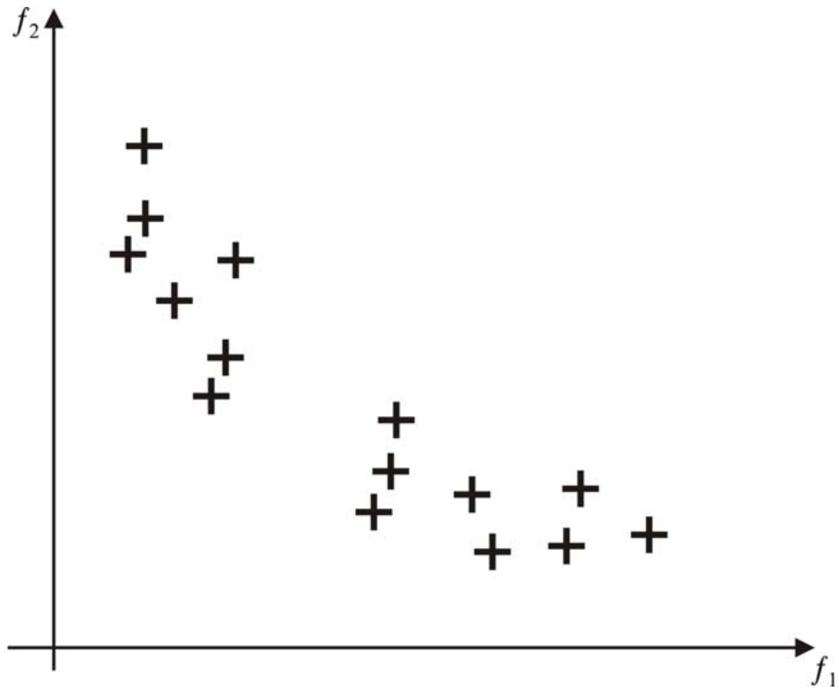


SMS(S-Metric Selection)-EMOA

State-of-the-art EMOA

$(\mu + 1)$-selection

1. non-dominated sorting
2. in case of incomparability: contributions to hypervolume of subset



Computational complexity of hypervolume

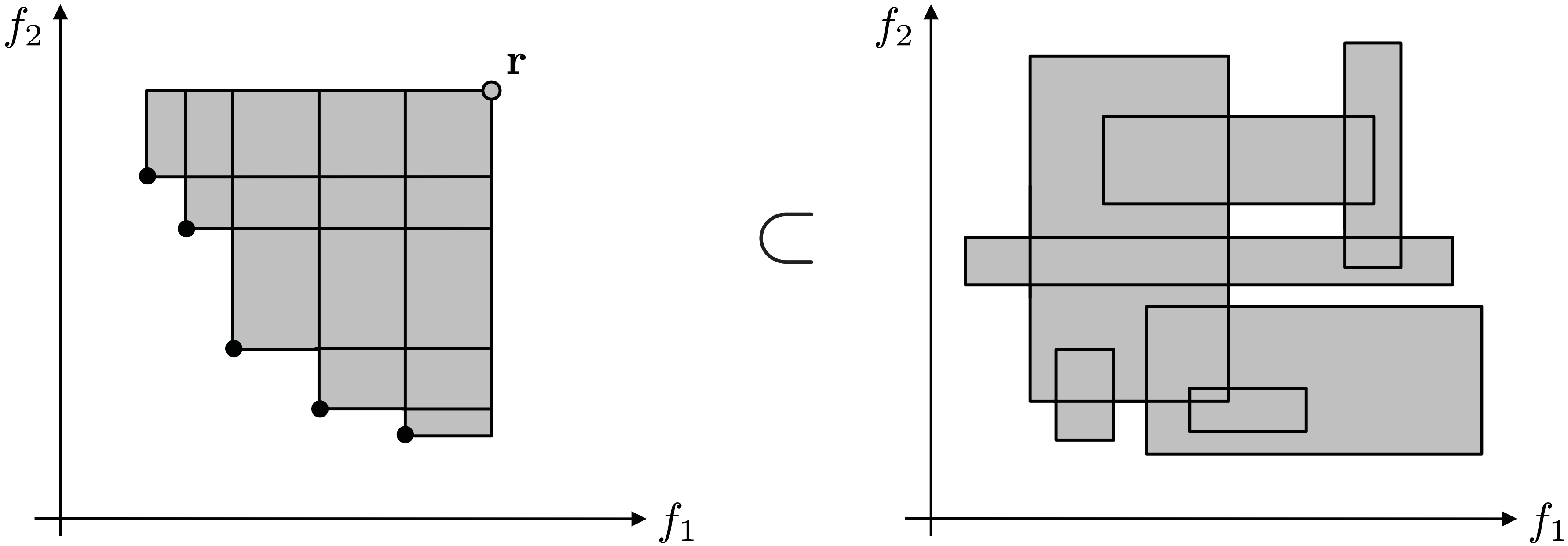
Lower Bound

$\Omega(m \log m)$

Upper Bound

$O(m^{d/2} \cdot 2^{O(\log^* m)})$

proof: hypervolume as special case of Klee's measure problem



Conclusions on EMOA

NSGA-II

only suitable in case of d=2 objective functions
otherwise no convergence to Pareto front

SMS-EMOA

also effective for $d > 2$ due to hypervolume
hypervolume calculation time-consuming
$\Rightarrow$use approximation of hypervolume

Other state-of-the-art EMOA, e.g.

- MO-CMA-ES: CMA-ES + hypervolume selection
- ϵ-MOEA: objective space partitioned into grid, only 1 point per cell
- MSOPS: selection acc. to ranks of different scalarizations

Conclusions

- real-world problems are often multiobjective
- Pareto dominance only a partial order
- a priory: parameterization difficult
- a posteriori: choose solution after knowing possible compromises
- state-of-the-art a posteriori methods: EMOA, MOEA
- EMOA require sortable population for selection
- use quality measures as secondary selection criterion
- hypervolume: excellent quality measure, but computationally intensive
- use state-of-the-art EMOA, other may fail completely