[center]

HEURISTIC OPTIMIZATION

Heuristic Algorithms for Multiobjective Combinatorial Optimization

Adapted from a tutorial by Luís Paquete given at SLS 2009

Introduction

Multiobjective Combinatorial Optimization Problems (MCOPs)

- Many real-life problems are multiobjective
  - Logistics and transportation
  - Timetabling and scheduling
  - ... and many others
- But most MCOPs are NP-hard and intractable

How to design and analyze SLS algorithms for MCOPs?

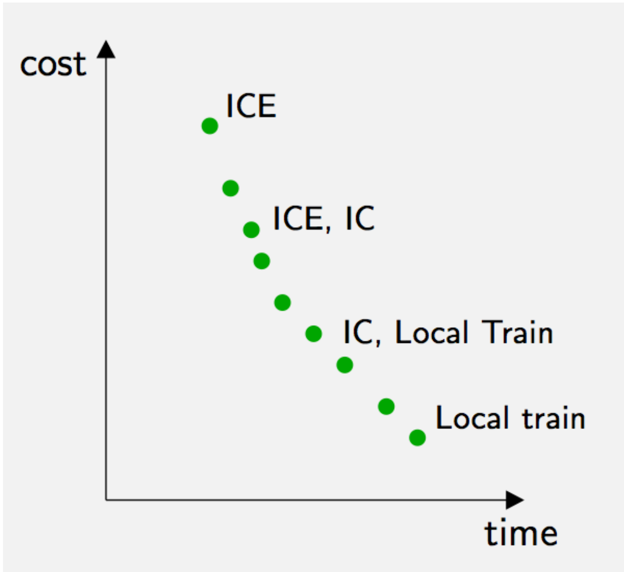
Train roundtrip through capitals of German federal states

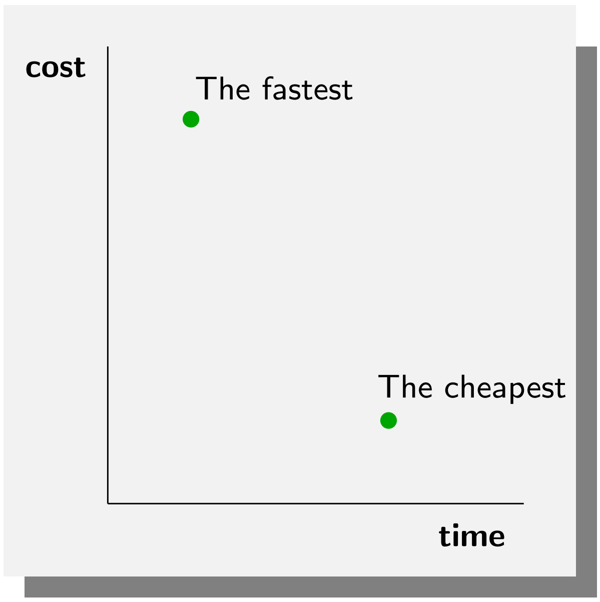
The fastest roundtrip:

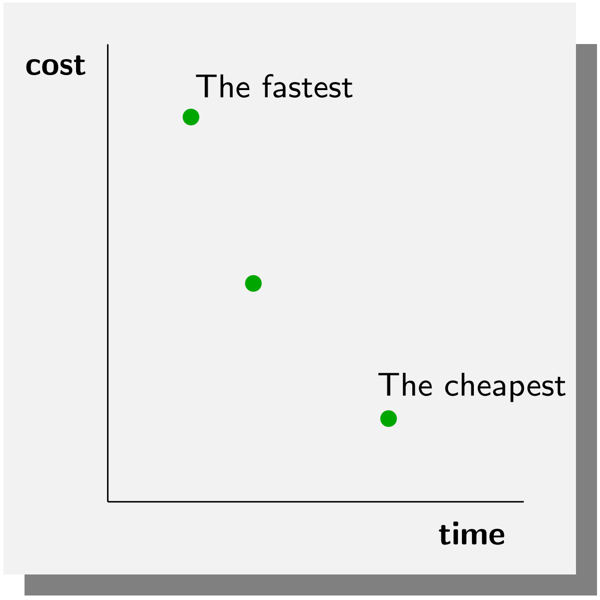
- take only ICE trains

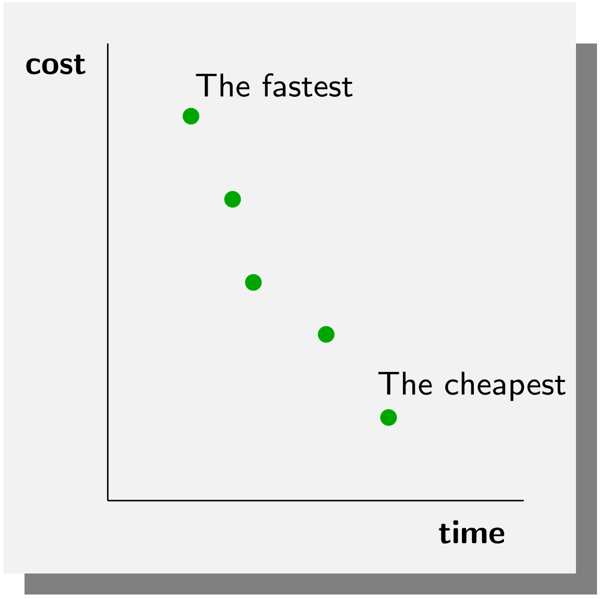
The cheapest roundtrip:

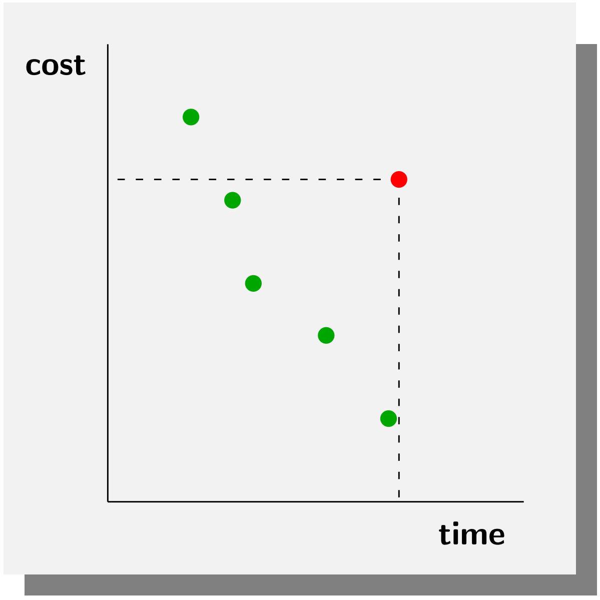
- take only local trains

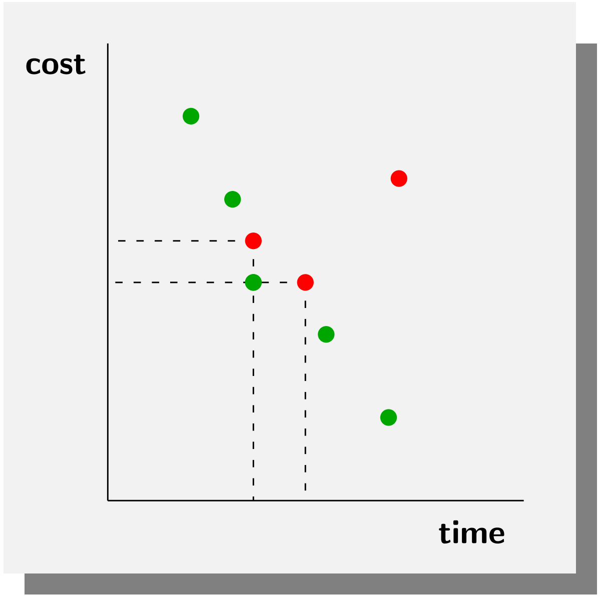





cost
The fastest
The cheapest
time


cost
The fastest
The cheapest
time


cost
time


cost
time

Multiobjective Combinatorial Optimization Problem

The set X of feasible solutions is finite and its elements have some combinatorial property (graph, tree, path, partition, etc.).

The goal is to

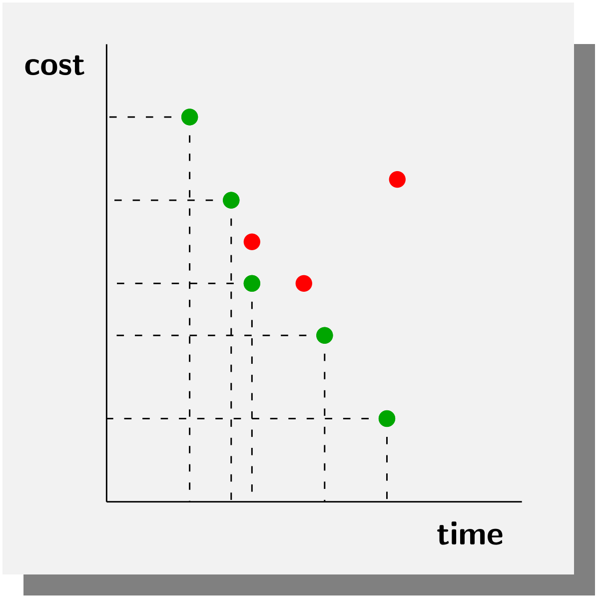
$$\min_{x \in X} \mathbf{f}(x) = (f_1(x), \dots, f_Q(x))$$

- The objective function $\mathbf{f}$ maps $x \in X$ to $\mathbb{R}^Q$

- Optimality depends of the decision maker's preferences (or lack of them).
- Pareto-optimality is based on component-wise order :

$$\mathbf{u} \leq \mathbf{v} \iff \mathbf{u} \neq \mathbf{v} \text{ and } u_i \leq v_i, \; i = 1, \dots, Q$$

- A solution $x \in X$ is efficient iff $\nexists \; x' \in X$ s.t. $\mathbf{f}(x') \leq \mathbf{f}(x)$
- Efficient set is the set of all efficient solutions
- Nondominated set is the image of the efficient set in $\mathbf{f}$



MCOPs and Solution Methods

- Most MCOPs are NP-hard

Decision version of MCOP (MCOP-D) [Serafini 1986]:
Given $\mathbf{z} = (z_1, \ldots, z_Q)$, does there exist a solution $x \in X$ s.t.

$$\mathbf{f}(x) \leq \mathbf{z} \quad \text{or} \quad \mathbf{f}(x) = \mathbf{z}?$$

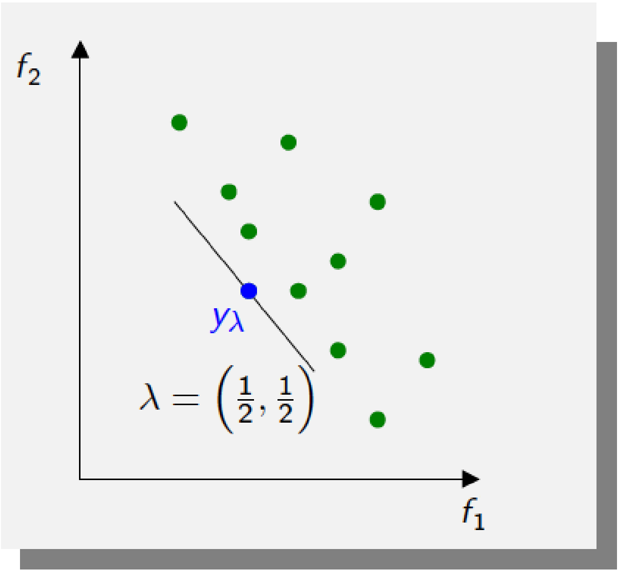
1. If the single-objective problem is NP-complete, then the corresponding MCOP-D is also NP-complete.
2. If the single-objective problem is solvable in polynomial time, the corresponding MCOP-D may still be NP-complete.

Solution Methods to MCOPs

- Enumeration Methods
  - Multiobjective Branch & Bound
  - Multiobjective Dynamic Programming
- Scalarized Methods
  - Solving several related single-objective problems
  - Weighted Sum, ϵ-constraint, etc.
- SLS Algorithms

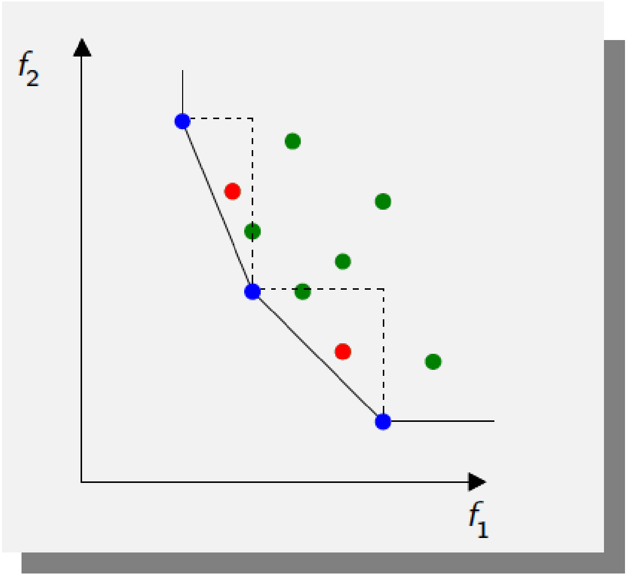
Weighted Sum

- $\min\limits_{x \in X} \sum_{i=1}^{Q} \lambda_i f_i(x)$
- λ gives a search direction
- An optimal solution with $\lambda > 0$ is efficient.



Weighted Sum

- $\min_{x \in X} \sum_{i=1}^{Q} \lambda_i f_i(x)$
- λ gives a search direction
- An optimal solution with $\lambda > 0$ is efficient.



SLS Algorithms

SLS Algorithm design challenges for MCOPs

- How to attain more than one solution?
- How to attain high quality solutions?
- How to evaluate performance?

Rule of thumb

- Closeness to the nondominated set
- Well-distributed outcomes
- The more, the better

Scalarized Acceptance Criterion (SAC) Model

- Weighted Sum

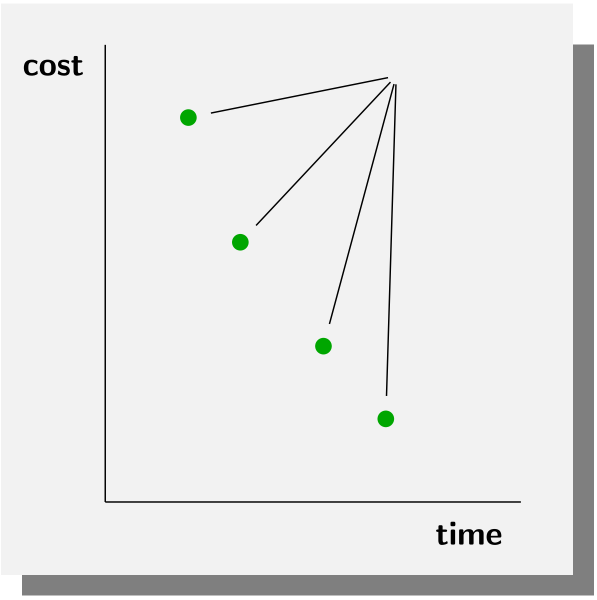
$$f(x) = \sum_{i=1}^{Q} \lambda_i f_i(x)$$

- Weighted Chebycheff

$$f(x) = \max_{i=1,\dots,Q} \left(\lambda_i \mid f_i(x) - y_i \mid\right)$$

SAC Search Model

input: weight vectors Λ
for each $\lambda \in \Lambda$ **do**
    x is a candidate solution
    $x' = \text{SolveSAC}(x, \lambda)$
    Add x' to Archive
Filter Archive
return Archive



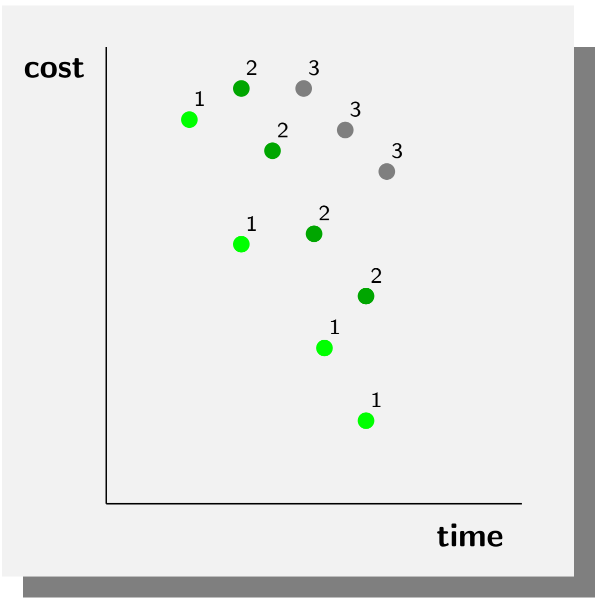
SAC Search Model

input: weight vectors Λ
for each $\lambda \in \Lambda$ **do**
  x is a candidate solution
  $x' = \text{SolveSAC}(x, \lambda)$
  Add x' to Archive
Filter Archive
return Archive

- Search Strategy
- Number of Scalarizations
- Intensification Mechanism
- Neighborhood

SAC Search Model – EMO

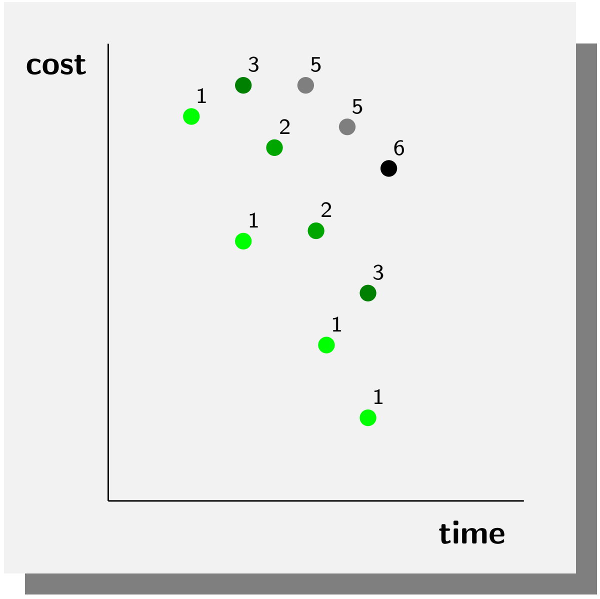
input: candidate solution set X_n
repeat
  $X_r = \text{Reproduce/Mutate}(X_n)$
  $R = \text{Rank}(X_r, X_n)$
  $X_s = \text{Select}(X_r, X_n, R)$
  $X_n = \text{Replace}(X_s)$
return X_n



SAC Search Model – EMO

---

input: candidate solution set X_n
repeat
  X_r = Reproduce/Mutate(X_n)
  R = Rank(X_r, X_n)
  X_s = Select(X_r, X_n, R)
  X_n = Replace(X_s)
return X_n



SAC Search Model – EMO

---

input: candidate solution set X_n
repeat
  X_r = Reproduce/Mutate(X_n)
  R = Rank(X_r, X_n)
  X_s = Select(X_r, X_n, R)
  X_n = Replace(X_s)
return X_n

- Component-wise order
- Closeness
- Performance indicators

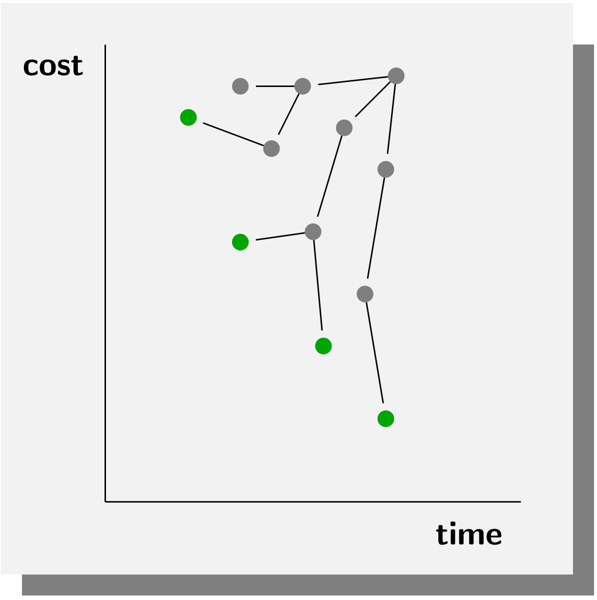
Multiobjective Local Search

input:candidate solution x
while x is not a local optimum **do**
  choose a neighbor x' from x such that $\mathbf{f}(x') \leq \mathbf{f}(x)$
  $x = x'$
return x

- What if $\mathbf{f}(x')$ and $\mathbf{f}(x)$ are mutually nondominated?
- How to obtain more than a single solution?

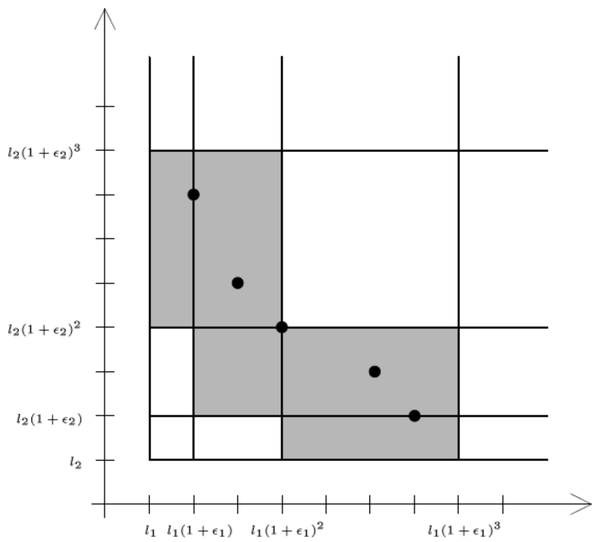
CWAC Search Model

input: candidate solution x
Add x to Archive
repeat
  Choose x from Archive
  X_N = Neighbors(x)
  Add X_N to Archive
  Filter Archive
until all x in Archive are *visited*
return Archive



CWAC Search Model

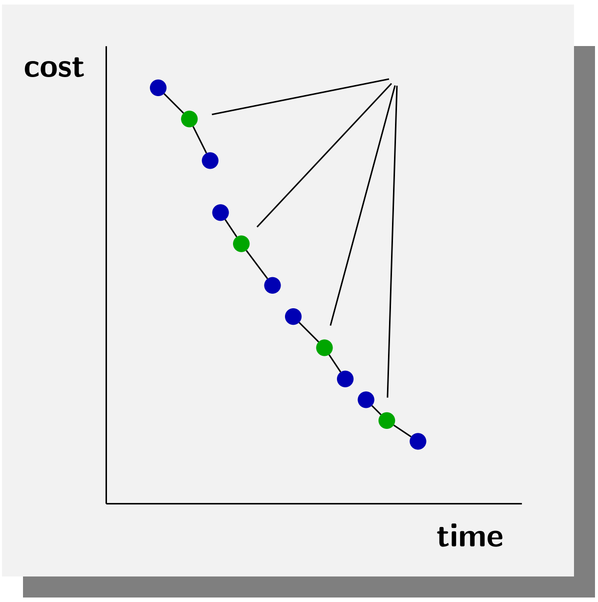
input: candidate solution x, ϵ
Add x to Archive
repeat
  Choose x from Archive
  X_N = Neighbors(x)
  Add X_N to Archive
  Filter Archive according to ϵ
until all x in Archive are *visited*
return Archive



Archive bounding
[Angel et al. 2004]

Hybrid Search Model

input: weight vectors Λ
for each $\lambda \in \Lambda$ **do**
  x is a candidate solution
  $x' =$ SolveSAC(x, λ)
  $X' =$ CW(x')
  Add X' to Archive
Filter Archive
return Archive



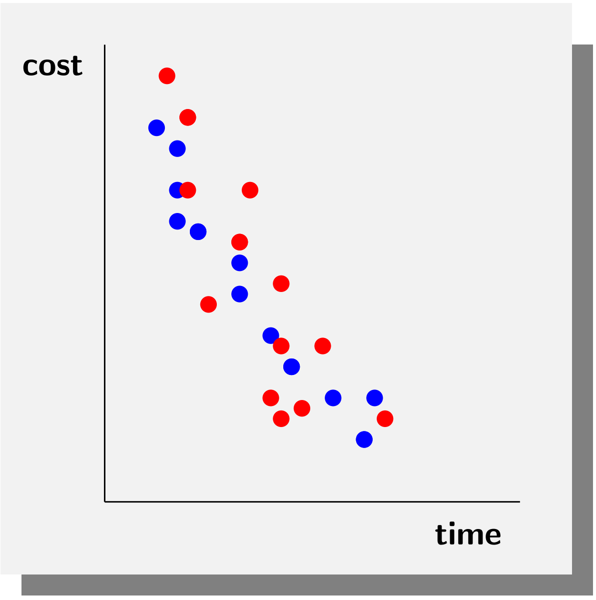
Performance Assessment

Rules of Thumb: An algorithm performs better if

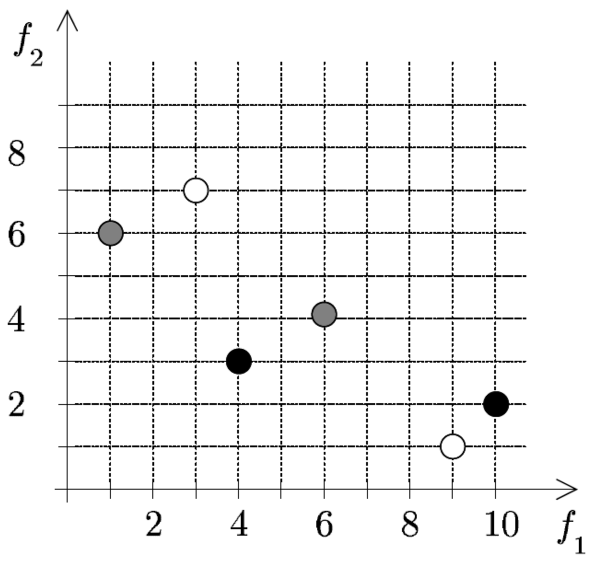
- It is closer to the nondominated set
- It has better distributed outcomes
- It has more solutions

Indicators of Performance

- Measure some property of the outcomes
- Most of the indicators have limitations [Knowles & Corne 2002, Zitzler et al. 2003]

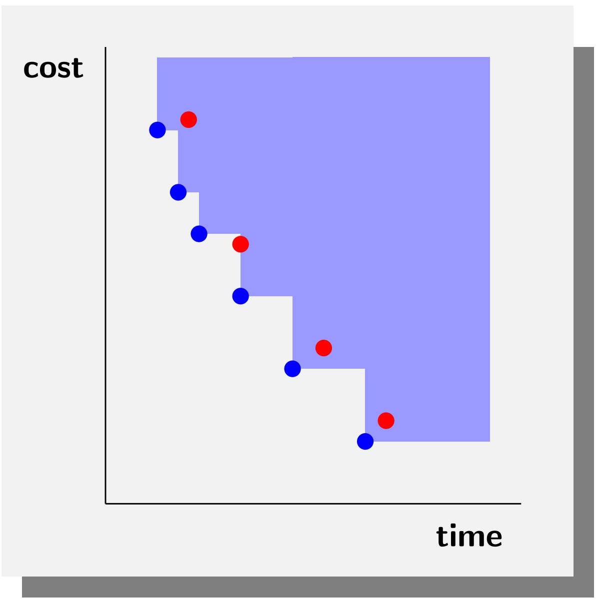


Many runs of Algorithms Blue and Red

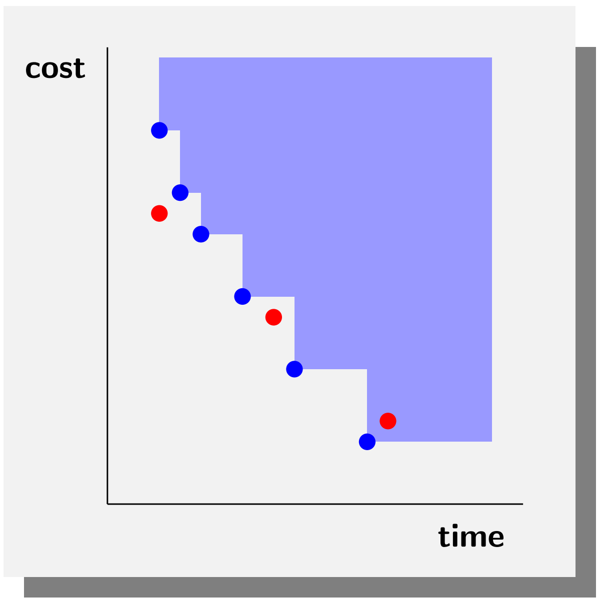


Another example

- Better relations
  [Hansen & Jaszkiewicz 1998, Zitzler et al. 2003]

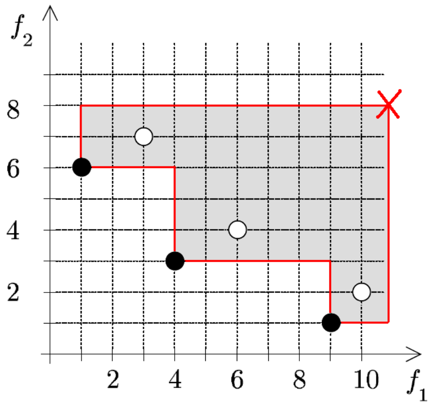


Blue is better than Red

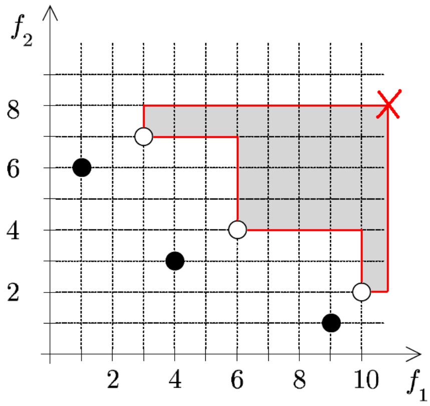


Blue and Red are incomparable

- Unary Indicator: Hypervolume
  [Zitzler and Thiele, 1998]



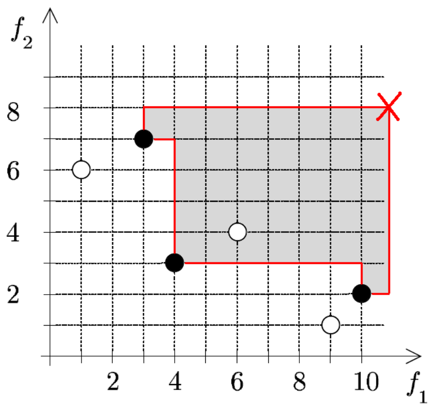
$H(B) = 45$



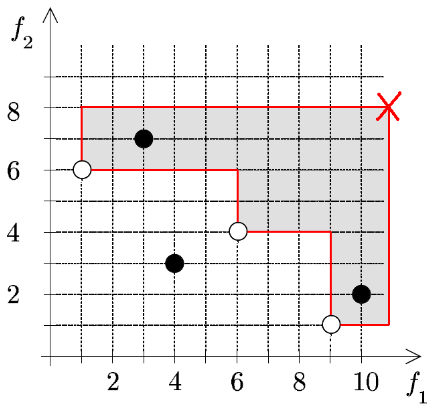
$H(W) = 25$

B is better than W $\implies$ $H(B) > H(W)$

- Unary Indicator: Hypervolume
  [Zitzler and Thiele, 1998]



$H(B) = 37$



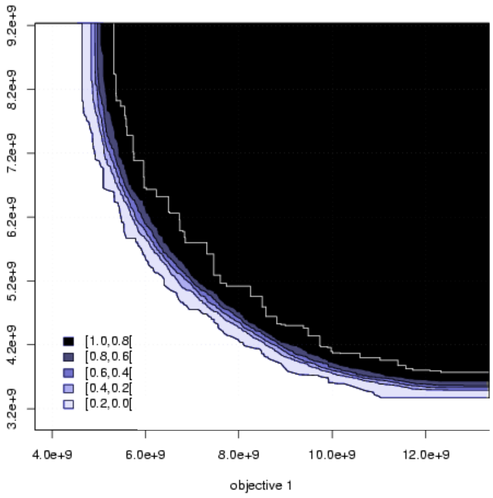
$H(W) = 36$

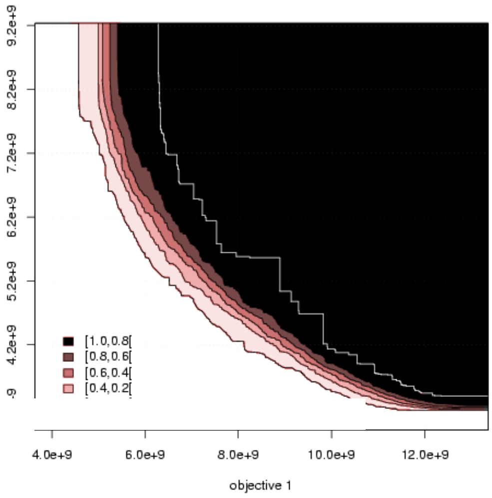
$H(B) > H(W)$ $\implies$ B is not worse than W

▶ Attainment Functions [V.G. da Fonseca et al. 2001]

AF: Prob. that an outcome set is better or equal to z.

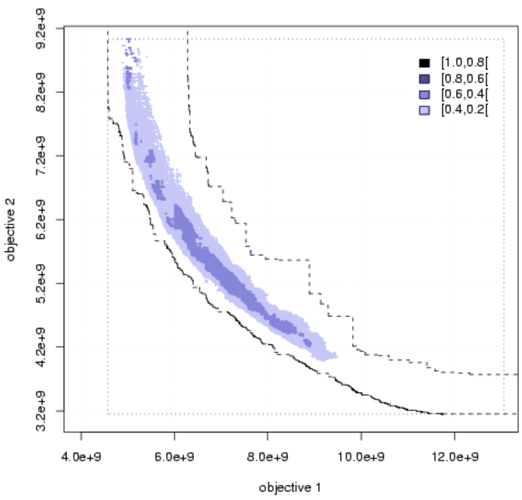
EAF: How many runs an outcome set is better or equal to z?

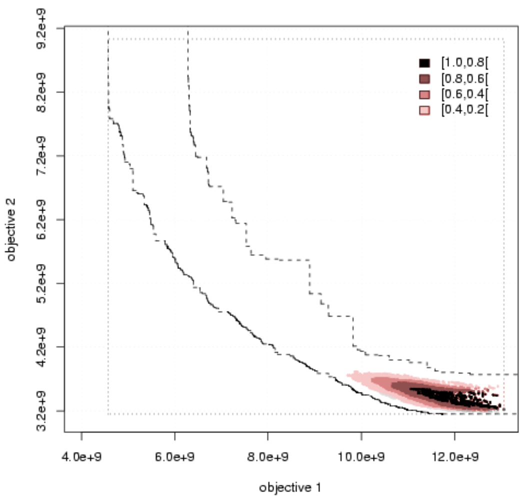




▶ Attainment functions – Visualization of differences

$$EAF_{Blue} - EAF_{Red}$$

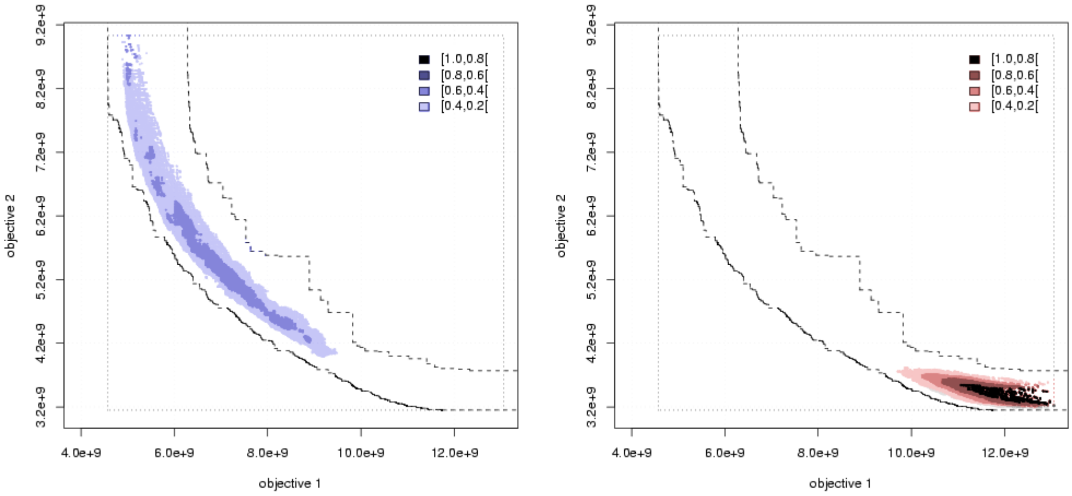




positive differences negative differences

- Attainment functions – Statistical testing

K-S test statistic: $\max \mid EAF_{Blue} - EAF_{Red} \mid$



positive differences

negative differences

References

- Textbooks: R.E. Steuer 1986, K. Miettinen 1999, M. Ehrgott 2005, V.T'kindt et al. 2002, K. Deb 2002.
- Reviews: M. Ehrgott and X. Gandibleux 2000, 2002, 2004, 2009, C.C. Coello 2000, D. Jones et al. 2002, J. Knowles and D. Corne 2004, L. Paquete and T. Stützle 2007.
- Complexity and Approximation: P. Hansen 1979, P. Serafini 1986, M. Ehrgott 2000, C.H. Papadimitriou and M. Yannakakis 2000, E. Angel et al. 2007.
- Performance Assessment: E. Zitzler et al. 2003, 2008, V.G. da Fonseca et al. 2001, 2010, M. Lopéz-Ibáñez et al. 2010.
- Web material: PISA (`http://www.tik.ethz.ch/~sop/pisa`), MOMH (`http://home.gna.org/momh`), ParadisEO (`http://paradiseo.gforge.inria.fr`)