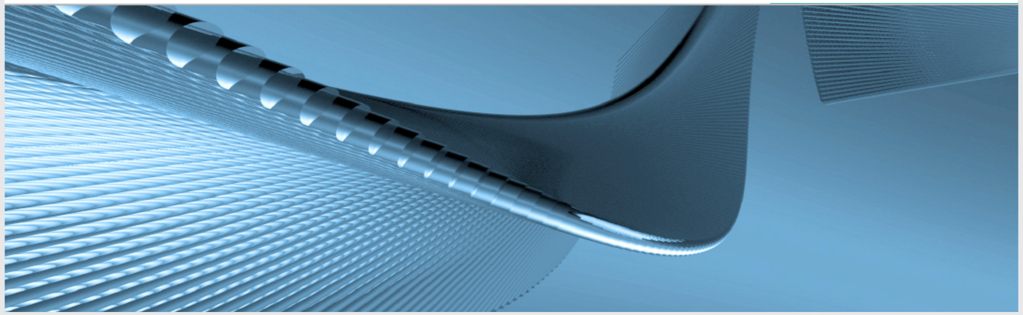
Multi-objective Optimization Inspired by Nature

Jürgen Branke
Institute AIFB
University of Karlsruhe, Germany
Karlsruhe Institute of Technology



Menu

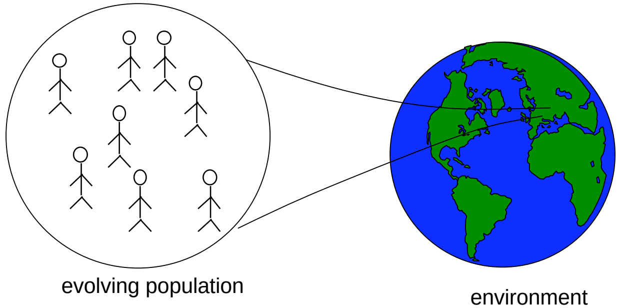
Appetizer
- Evolutionary algorithms

Main course
- Using evolutionary algorithms instead of exact optimizers for MOPs
- Multi-objective evolutionary algorithms
- Including preference information in MOEAs

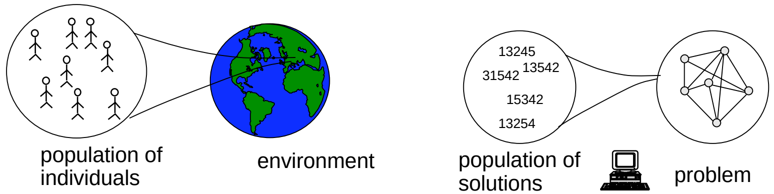
Desert
- Current research
- Summary

Evolutionary algorithms

Darwin's principle of natural evolution:
survival of the fittest
in populations of individuals (plants, animals), the better the individual is adapted to the environment, the higher its chance for survival and reproduction.



Transfer to optimization



Natural evolution
- individual
- environment
- fitness/how well adapted
- survival of the fittest
- mutation
- crossover

Evolutionary algorithms
- potential solution
- problem
- cost/quality of solution
- good solutions are kept
- small, random perturbations
- recombination of partial solutions

Basic algorithm

INITIALIZE population
(set of solutions)

EVALUATE Individuals
according to goal ("*fitness*")

REPEAT

SELECT parents

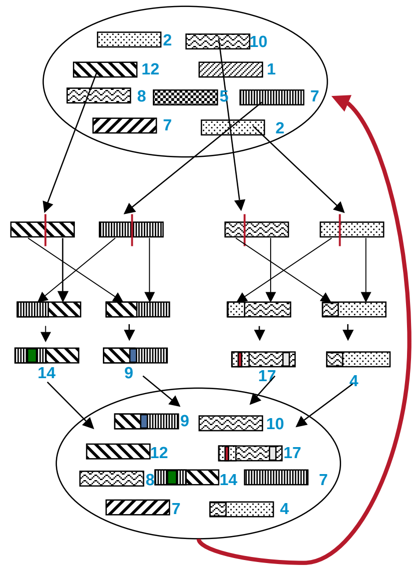
RECOMBINE parents (CROSSOVER)

MUTATE offspring

EVALUATE offspring

FORM next population

UNTIL termination-condition



Advantages/Disadvantages

+ No restriction w.r.t. fitness function (e.g. does not have to be differentiable)
+ Universal applicability
+ Easy to integrate heuristic knowledge if available
+ Easy to parallelize
+ Easy to use (usually inexpensive to develop)
+ Anytime algorithms (available time is fully utilized)
+ Can deal with multiple objectives
+ User-interaction possible
+ Allow for continuous adaptation
+ Can work with stochastic fitness functions

- Computationally expensive
- No guaranteed solution quality
- Parameter tuning necessary

Industrial applications

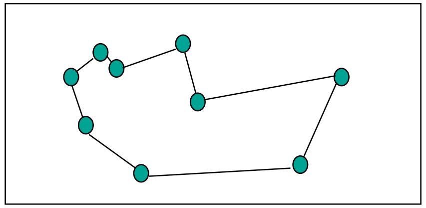
- Warehouse location problem (Locom)
- Process scheduling (Unilever)
- Job shop scheduling (Deer & Company, SAP, Volvo)
- Turbine design (Rolce Royce, Honda)
- Portfolio optimization (First Quadrant)
- Cleaning team assignment (Die Bahn)
- Chip design (Texas Instruments)
- Roboter movement (Honda)
- Nuclear fuel reloading (Siemens)
- Design of telephone networks (US West)
- Games (creatures)
- Military pilot training (British Air Force)
- Vehicle routing (Pina Petroli)
- Coating of fuorescent lamps (Philips)
- ....

Major design decisions

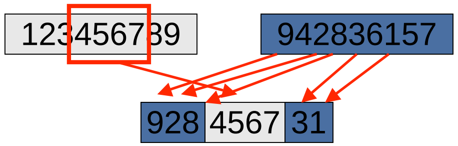
- Representation
- Genetic operators

- Selection mechanism
- Crossover/Mutation probability
- Population size
- Stopping criterion

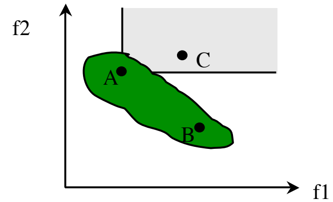
Simple example: Travelling Salesman Problem



- Permutation encoding: 3-1-4-5-7-2-6-8-9
- Mutation: Exchange two cities
- Order crossover (OX)
  - select partial sequence from one parent, fill up in order of other parent



Multiple objectives



- It is not always clear which solution is better
- Let f_i, $i=1\ldots d$ be the different optimization criteria. Then,
  a solution x is said to **dominate** a solution y ($x \succ y$)if and only if the following condition is fulfilled:

$$x \succ y \Leftrightarrow \quad f_i(x) \leq f_i(y) \quad \forall i \in \{1..d\}$$
$$\wedge \exists j : f_j(x) < f_j(y)$$

- Among a set of solutions P, the **non-dominated set** of solutions P' are those that are not dominated by any member of the set P
- A solution which is not dominated by any other solution in the search space is called **Pareto-optimal**.
- User preferences are required

Menu

Appetizer

Evolutionary algorithms

Main course

Using evolutionary algorithms instead of exact optimizers for MOPs

Multi-objective evolutionary algorithms

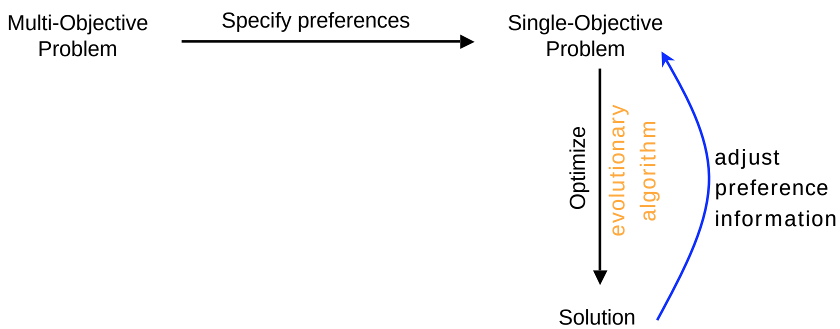
Including preference information in MOEAs

Desert

Current research

Summary

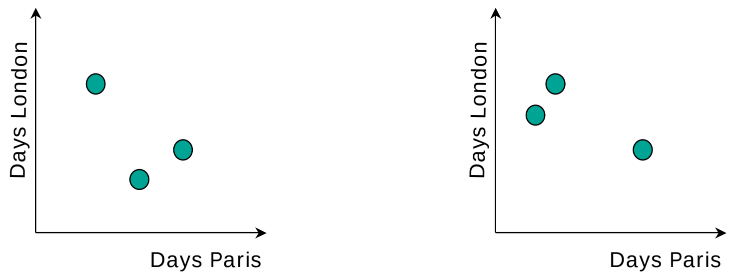
A priori approach



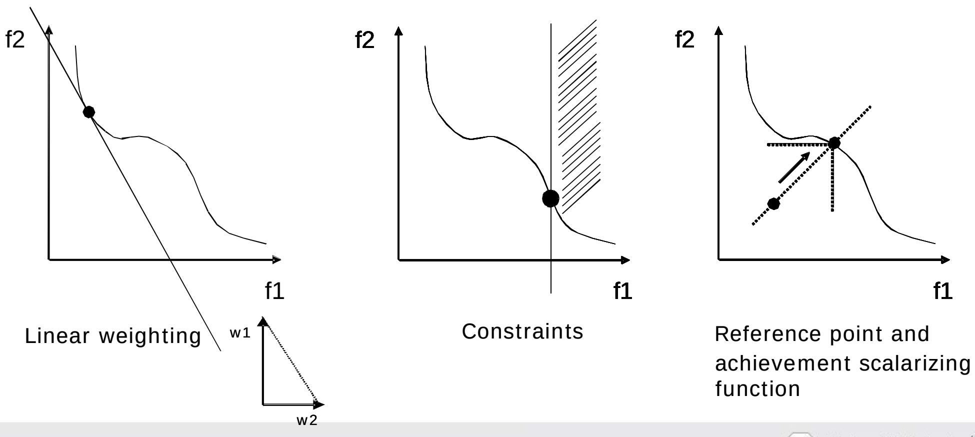
Specifying preferences

Difficult!

Example: Tell me which travel plan you prefer!



Specifying preferences



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

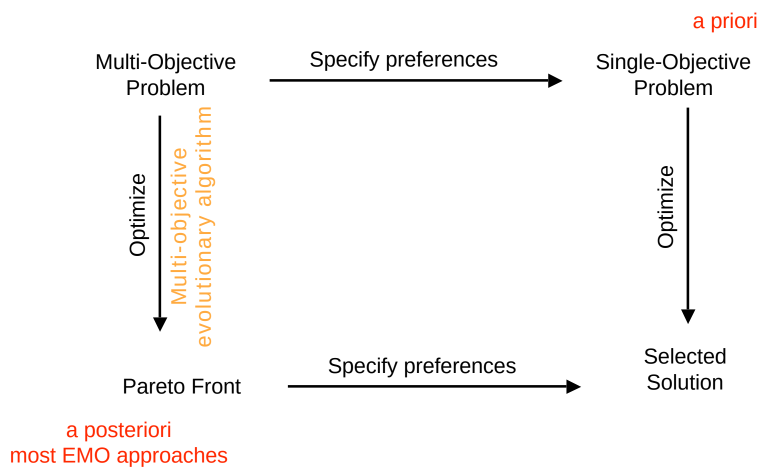
Advantages / Disadvantages of EAs

+ Allows to solve problems where no exact methods exist

– Metaheuristics do not guarantee (Pareto-) optimality
– Solutions generated in subsequent iterations may dominate each other
– Adjusting preference information may lead to unexpected results
– Computationally expensive

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

A posteriori - The power of populations

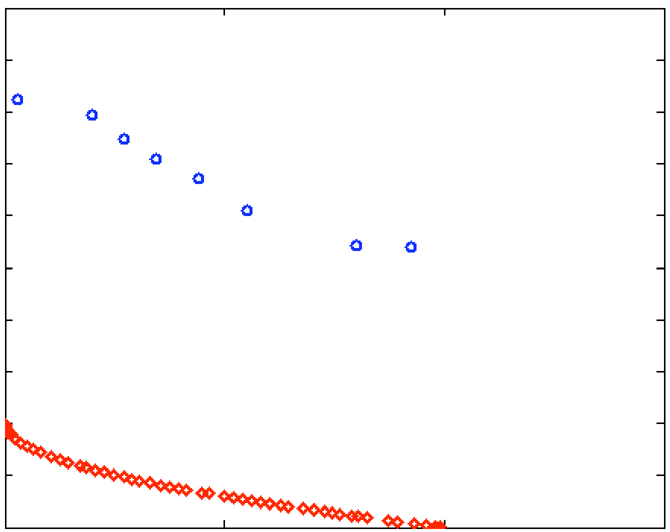


Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Multiobjective Evolutionary Algorithms (MOEAs)

- Since EAs work with a population of solutions, they can search for all (a representative subset of) Pareto-optimal solutions in one run
- Single EMO run is usually much more effective than multiple runs with different objectives

Demo



Basic structure

INITIALIZE population
(set of solutions)
EVALUATE Individuals
according to goal ("*fitness*")

REPEAT

SELECT parents

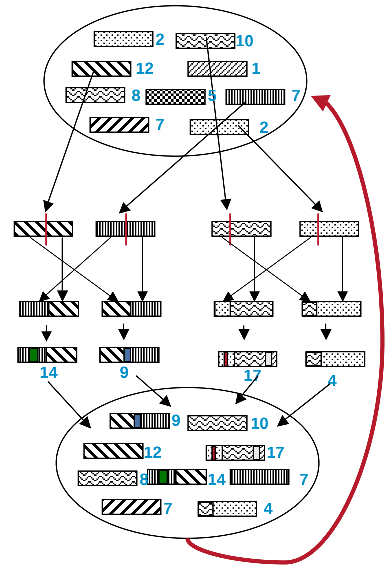
RECOMBINE parents (CROSSOVER)

MUTATE offspring

EVALUATE offspring

FORM next population

UNTIL termination-condition

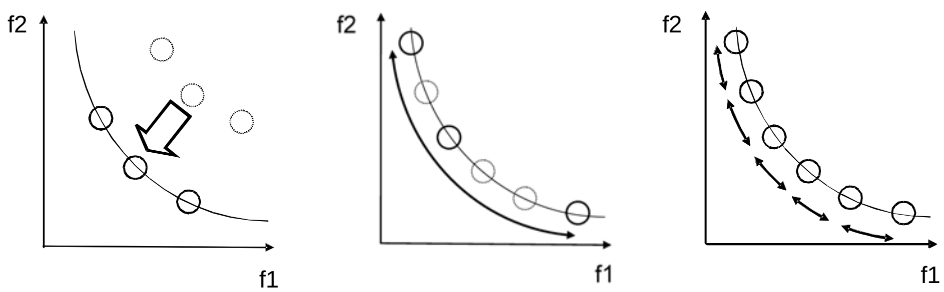


What is a good approximation?

close to the Pareto front

wide spread

equally distributed



Non-dominated Sorting GA (NSGA-II) [Deb et al. 2002]

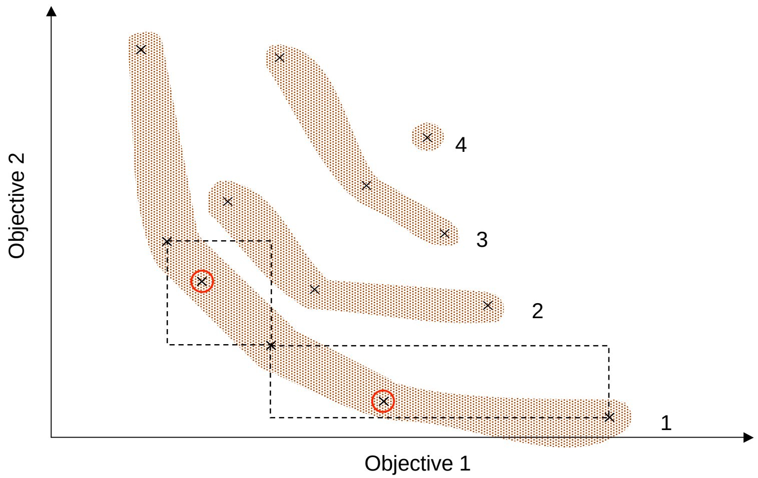
Based on two ideas:

1. **Pareto ranking:** based on Pareto-dominance
2. **Crowding distance**: mechanism to maintain diversity in the population

Other popular approach:

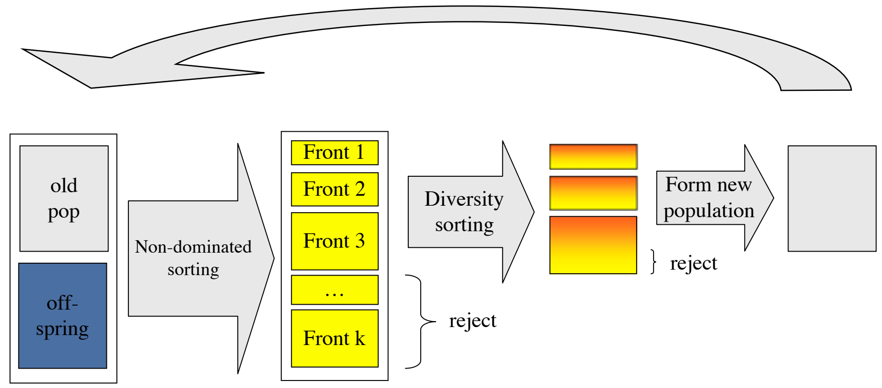
Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler

Pareto ranking + crowding distance



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

NSGA-II: Overall algorithm



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Advantages of finding the complete front

+ Not necessary to specify preferences a priori
+ Allows DM to choose solution after having seen the alternatives

+ Interactive search of Pareto front
  - Optimization prior to interaction, thus interaction very fast
  - Only non-dominated solutions are presented to the user
  - Direct navigation by user is possible
  - Additional information on distribution of solutions along the front may be provided to the user (nadir point, ideal point, ...)

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Advantages of finding the complete front

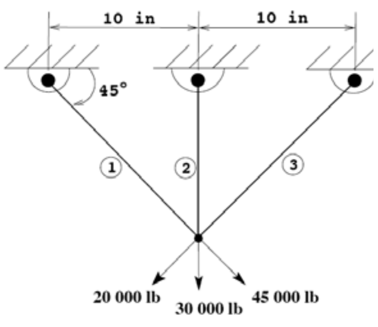
- + Not necessary to specify preferences a priori
- + Allows DM to choose solution after having seen the alternatives

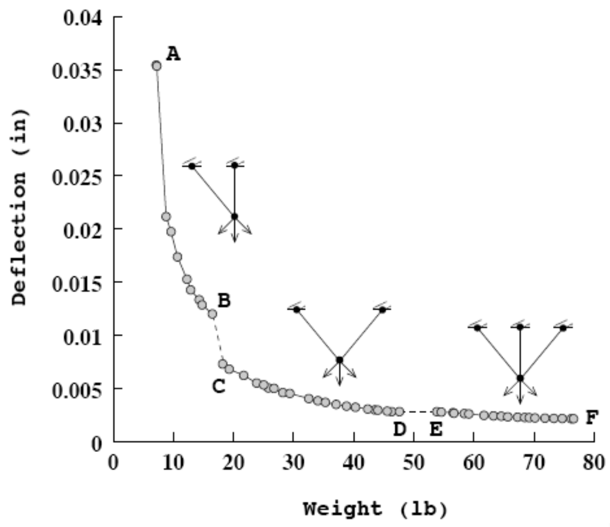
- + Interactive search of Pareto front

- + Offer different alternatives to different customers (e.g., mean-variance portfolio optimization)

- + Reveal common properties among Pareto-optimal solutions (some variables are always the same)
- + Understand the causes for the trade-off

Understanding trade-offs





Advantages of finding the complete front

- + Allows DM to choose solution after having seen the alternatives
- + Interactive search of Pareto front

- + Offer different alternatives to different customers (e.g., mean-variance portfolio optimization)

- + Reveal common properties among Pareto-optimal solutions (some variables are always the same)
- + Understand the causes for the trade-off

- + Aid in other optimization tasks (constraints, multi-objectivization)

Do we really need the whole front?

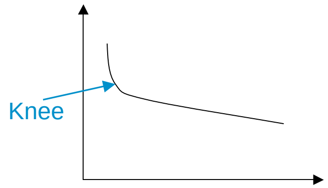
- Computational overhead
- Large set of alternatives, difficult to search by DM

- Identify “most interesting” regions
- Take into account partial user preferences

- Bias the distribution
- Restrict the distribution

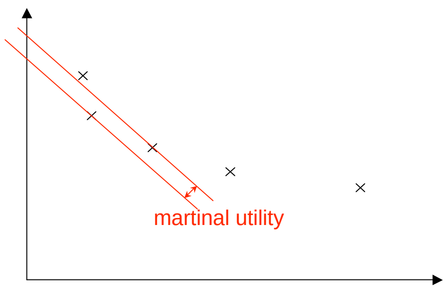
Identifying knees

- Solutions where an improvement in either objective leads to a significant worsening of the other objective are more likely to be preferred -> knee

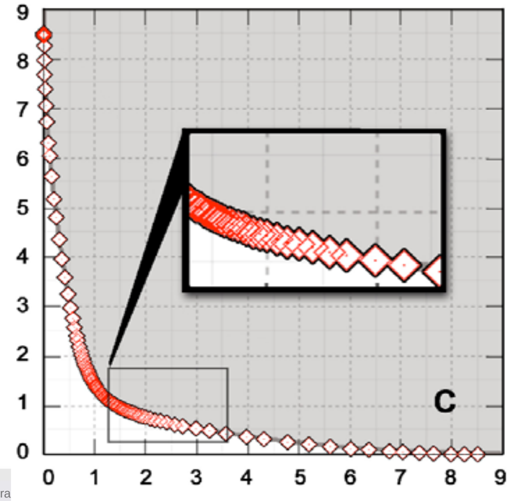


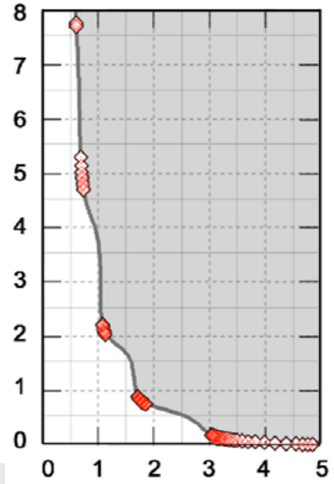
Marginal Utilities [Branke et al. 2004]

- Assume user has linear utility function
- Evaluate each solution with expected loss of utility if solution would not be there



- If crowding distance is replaced by marginal utilities, algorithm focuses on knees
- No preference information necessary





Menu

Appetizer
- Evolutionary algorithms

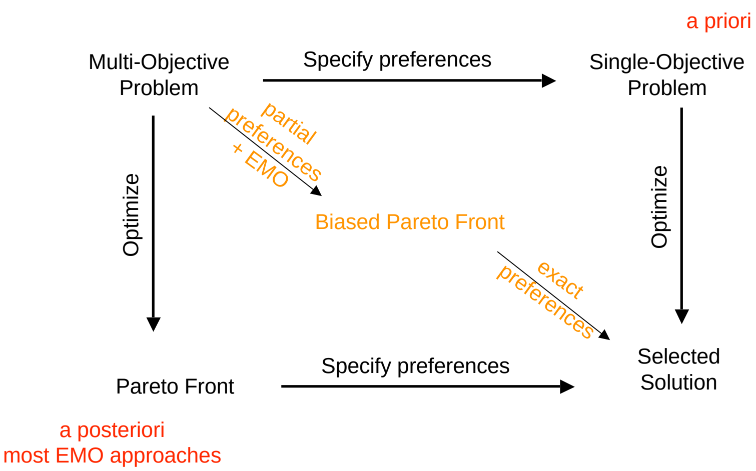
Main course
- Using evolutionary algorithms instead of exact optimizers for MOPs
- Multi-objective evolutionary algorithms
- Including preference information in MOEAs

Desert
- Current research
- Summary

Motivation

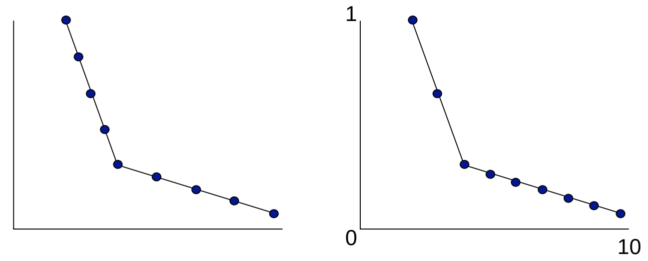
- Although a user generally cannot specify his/her preferences exactly before alternatives are known, he/she usually has some rough idea as to what solutions are desired
  - "A solution should have at least x in objective f1."
  - "f1 of x would be good, f1 of y would be great."
  - "My target solution would look something like this."
  - "If a solution is worse by one unit in objective f1, it should be at least x units better in objective f2 to be interesting."
  - "Objective f1 is somewhat more important than objective f2."
- Hope: Find a larger variety of more interesting solutions more quickly.

Considering partial preferences



EMO doesn't need user preferences. Does it?

- All EMO approaches attempt to find a representative set of the Pareto optimal front
- Usually, representative means well distributed



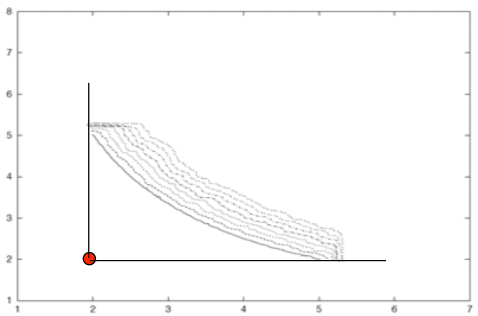
- But: distribution depends on scaling of the objectives

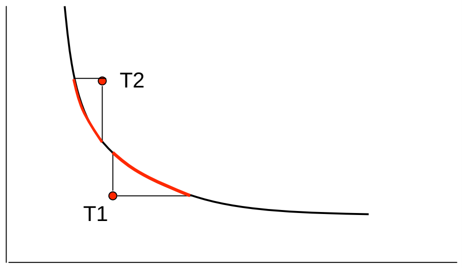
"... at least x in objective f1."

- Constraint: f1>x
- Constraints are easy to integrate into EMO
  - Lexicographic ordering (feasible solution always dominates infeasible solution)
  - Penalty
  - Additional objective
  - ...

"... target solution ..."

1. Minimize distance to ideal solution (single objective)
2. Minimize maximal distance in any objective (single objective)
3. Goal Attainment [Fonseca & Fleming, 1993]/Goal Programming [Deb, 1999]
   - Do not reward improvement over ideal solution
     f1 -> max{0, f1-f1*}



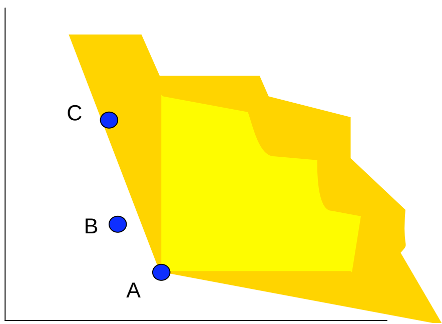


"... at least x units better in objective f2 ..."

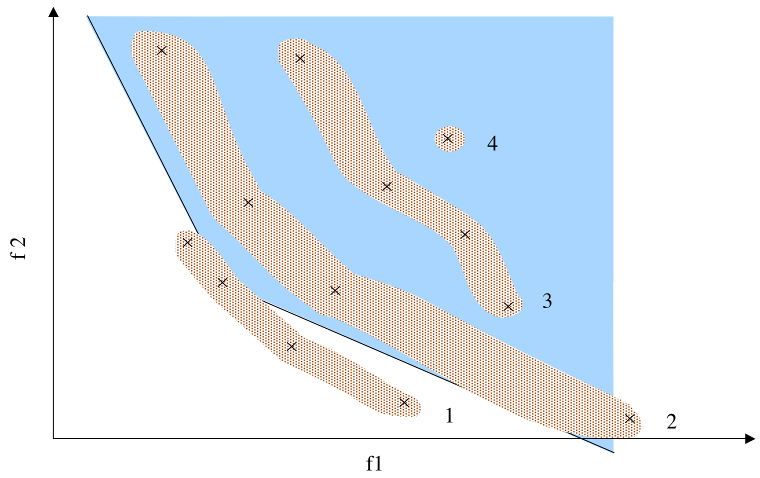
- Maximal and minimal trade-offs
- Guided MOEA [Branke et al. 2001]
- Modify definition of dominance

- Can be achieved by a simple transformation of the objectives
- Not so easy for more than 2 objectives

$$\Omega_1(x) = f_1(x) + w_{12} f_2(x)$$
$$\Omega_2(x) = f_2(x) + w_{21} f_1(x)$$
$$x \succ y \Leftrightarrow \quad \Omega_i(x) \le \Omega_i(y) \ \forall i \in \{1,2\}$$
$$\wedge \exists j : \Omega_j(x) < \Omega_j(y)$$

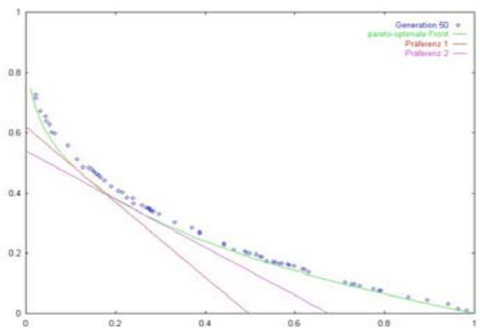


Guided dominance criterion

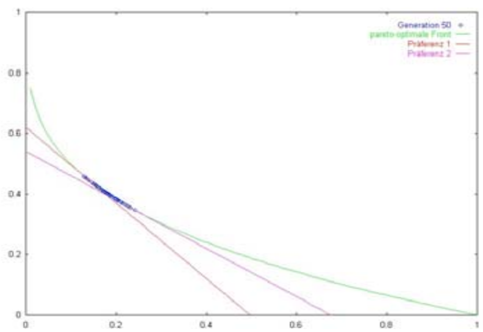


The effect of guidance

standard MOEA



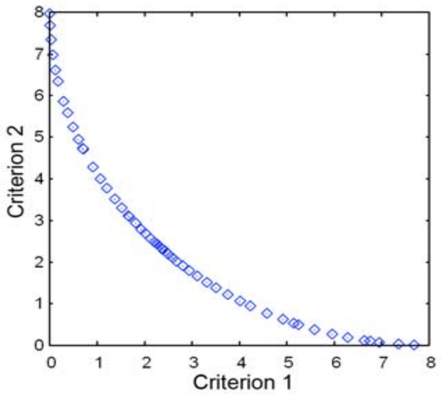
guided MOEA

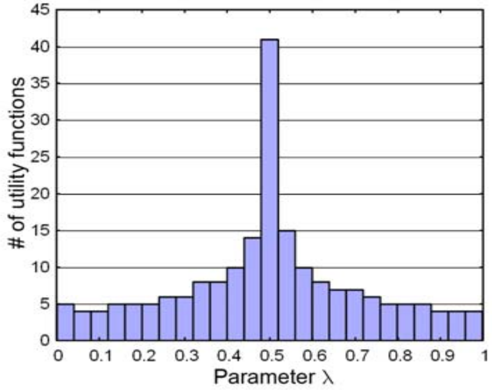


⇨ Faster convergence and better coverage of the interesting area of the Pareto-optimal front

Marginal utility with preferences

- With non-uniform distribution of utility functions

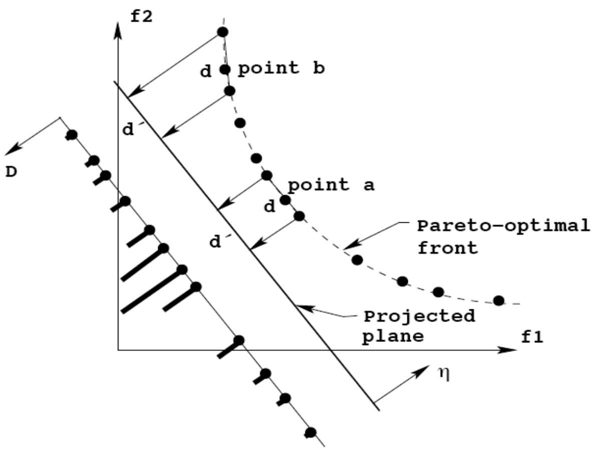


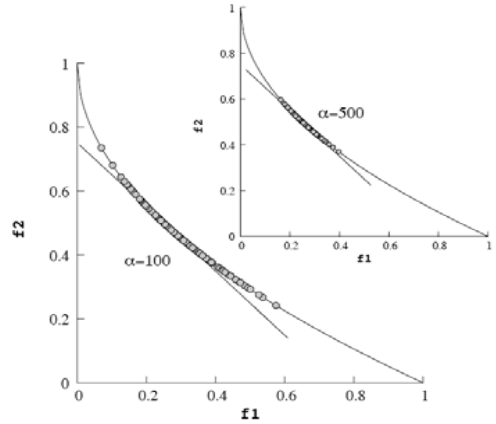


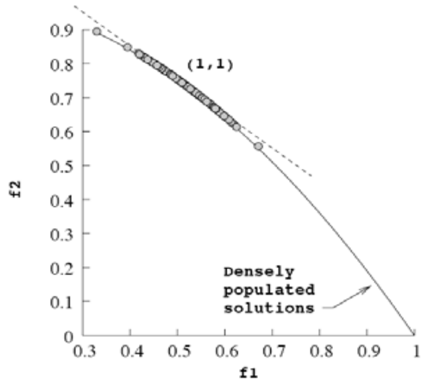
Biased sharing [Branke & Deb 2004]

- User specifies weights and spread parameter
- Crowding distance calculation is modified

$$D_i = d_i \left(\frac{d_i'}{d_i}\right)^{\alpha}$$

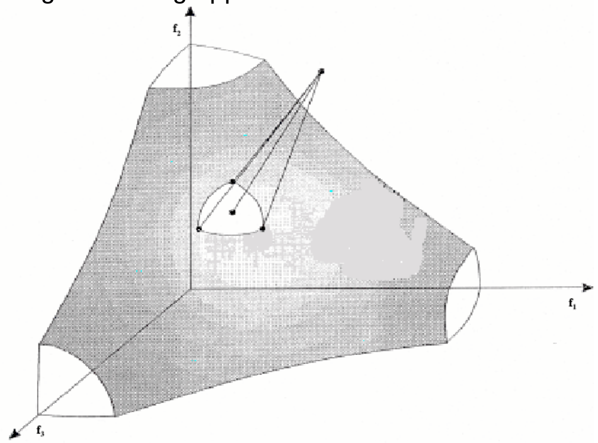






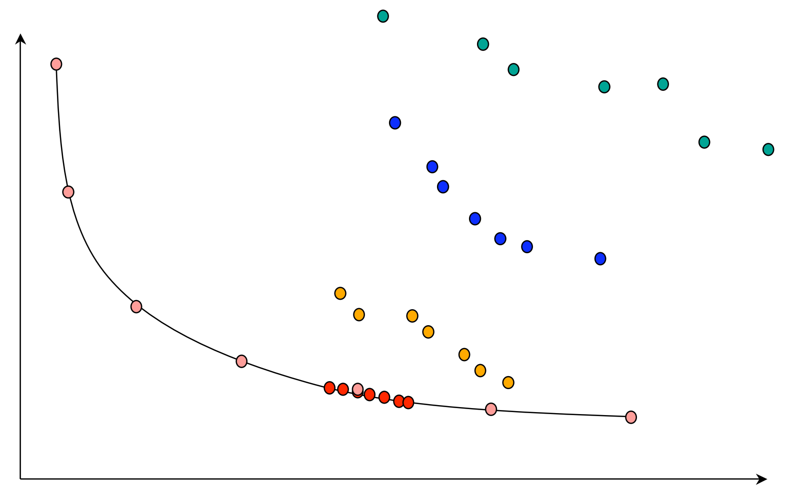
Light Beam Search [Deb&Kumar 2007]

- Specify aspiration and reservation point
- Determine projection on Pareto front
- Identify interesting local area using outranking approaches



Interactive MOEA

- Narrow down / refocus search during MOEA run
- Explicitly by
  - adjusting constraints
  - moving the target
  - modifying the max/min trade-offs
  - ...
- Implicitly by comparing solutions
  - learn user preferences



Menu

Appetizer
- Evolutionary algorithms

Main course
- Using evolutionary algorithms instead of exact optimizers for MOPs
- Multi-objective evolutionary algorithms
- Including preference information in MOEAs

Desert
- Current research
- Summary

Current research

- Many-objective problems (difficulty: almost all solutions non-dominated)
- Multiobjectivization (influence diversity and search space structure)
- Noisy objective functions (e.g., stochastic simulation)
- Worst-case multi-objective optimization
- Using metamodels in case of expensive fitness evaluation
- Individual = Set of solutions

Summary

- Evolutionary algorithms open new possibilities in multi-objective optimization because
  - they are very general problem solvers
  - they work on a population of solutions and can thus search for a whole set of solutions simultaneously
- Different ways to use EAs in MOO:
  1. As single-objective optimizer in classical MOO techniques
  2. To generate an approximation to the whole Pareto front
  3. With partial user preferences resulting in a partial front or biased distribution
  4. Interactively guided by the user

EMO resources

Books:
- K. Deb: "Multi-objective optimization using evolutionary algorithms". Wiley, 2001
- C. Coello Coello, D. A. Van Veldhuizen and G. B. Lamont: "Evolutionary algorithms for solving multi-objective problems". Kluwer, 2002
- J. Branke, K. Deb, K. Miettinen, R. Slowinski: "Multi-objective optimization - interactive and evolutionary approaches". Springer, to appear

Websites:
- http://www.lania.mx/~coello

References

[Branke et al. 2001] J. Branke, T. Kaußler, H. Schmeck: "Guidance in evolutionary multi-objective optimization". Advances in Engineering Software, 32:499-507

[Branke et al. 2004] J. Branke, K. Deb, H. Dierolf, M. Osswald: "Finding knees in multi-objective optimization". Parallel Problem Solving from Nature, Springer, pp. 722-731 Branke and Deb Biased

[Branke and Deb 2004] J. Branke and K. Deb: "Integrating user preferences into evolutionary multi-objective optimization". In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, Springer, pages 461–477

[Deb 1999]: "Solving goal programming problems using multi-objective genetic algorithms". Congress on Evolutionary Computation, IEEE, pp. 77-84

[Deb et al. 2002] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan: "A fast and Elitist multi-objective Genetic Algorithm: NSGA-II". IEEE Transactions on Evolutionary Computation 6(2):182-197

[Deb and Kumar 2007] K. Deb and A. Kumar: "Light beam search based multi-objective optimization using evolutionary algorithms". Congress on Evolutionary Computation, IEEE, pp. 2125-2132

[Fonseca and Fleming 1993] C. M. Fonseca and P. J. Fleming: "Genetic algorithms for multiobjective optimization: Formulation, discussion, and generatization". International Conference on Genetic Algorithms, Morgan Kaufmann, pp. 416-423