Tutorial on Evolutionary Multiobjective Optimization GECCO 2006

Eckart Zitzler, Stefan Bleuler

Computer Engineering (TIK), ETH Zurich, Switzerland

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Laboratory

Introductory Example: The Knapsack Problem

weight = 750g	weight = 1500g	weight = 300g	weight = 1000g
profit = 5	profit = 8	profit = 7	profit = 3

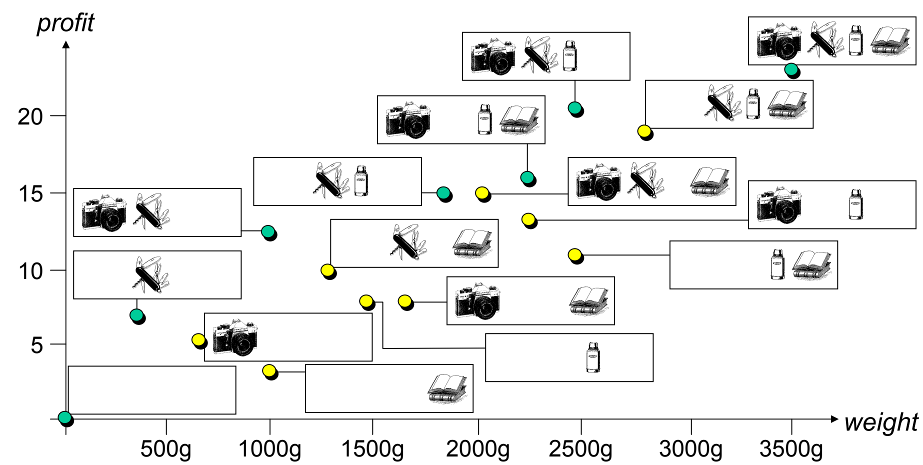
Single objective:

choose subset that
- maximizes overall profit
- w.r.t. a weight limit

Multiobjective:

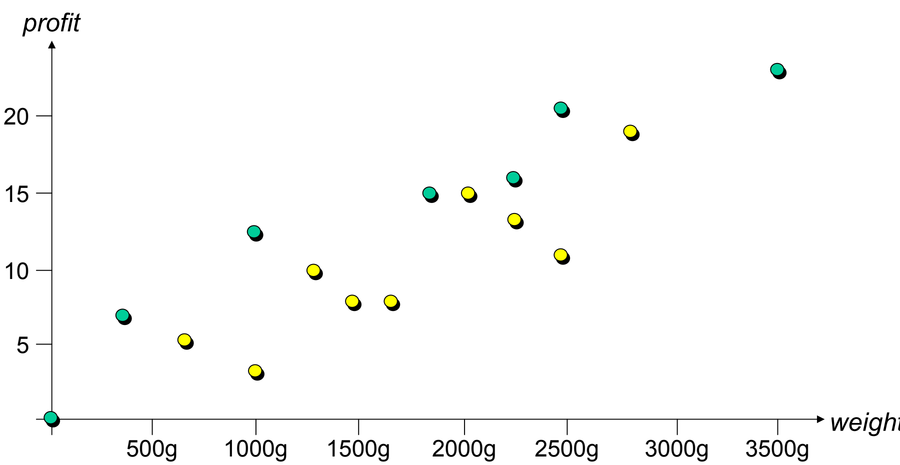
choose subset that
- maximizes overall profit
- minimizes overall weight

The Search Space



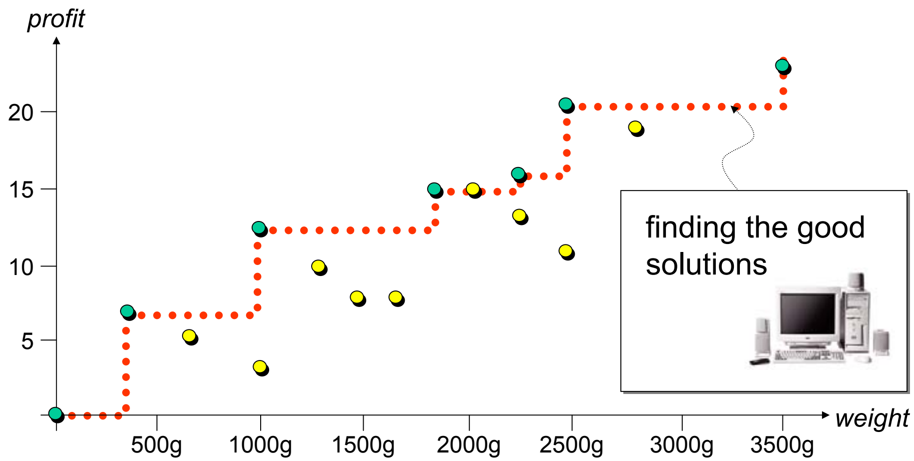
The Trade-off Front

Observations:
❶ there is no single optimal solution, but
❷ some solutions (●) are better than others (●)



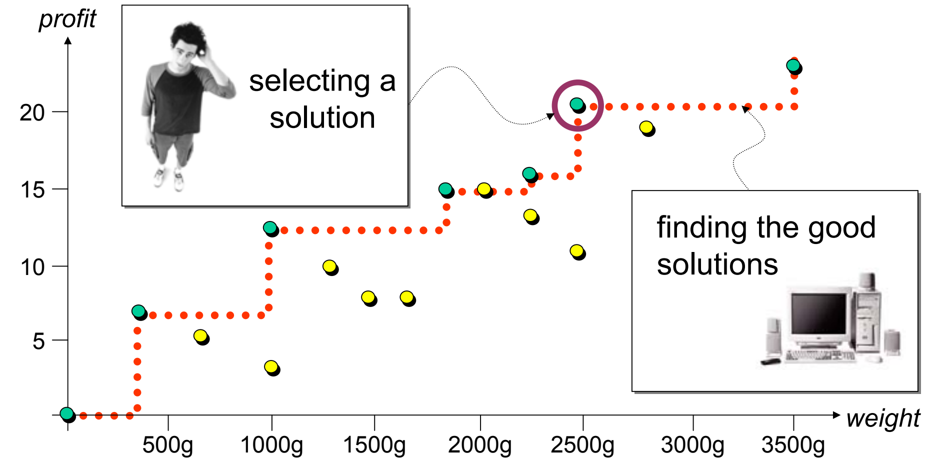
The Trade-off Front

Observations: ❶ there is no single optimal solution, but
❷ some solutions (●) are better than others (●)



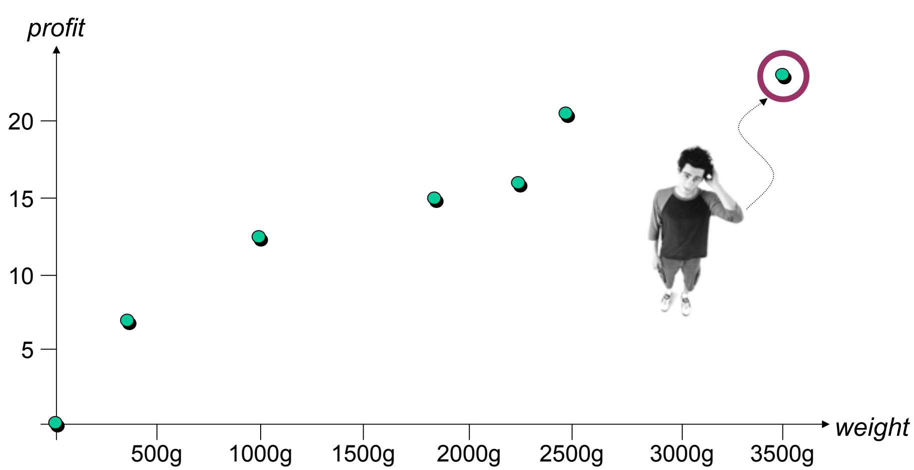
The Trade-off Front

Observations: ❶ there is no single optimal solution, but
❷ some solutions (●) are better than others (●)



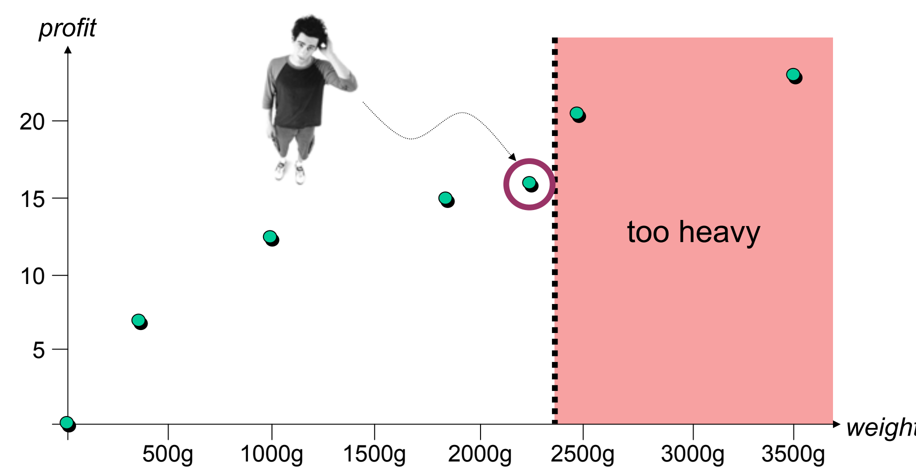
Decision Making: Selecting a Solution

Approaches:
- profit more important than cost (ranking)



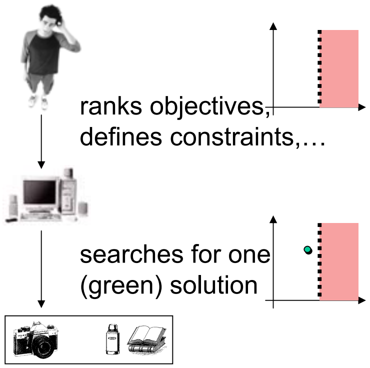
Decision Making: Selecting a Solution

Approaches:
- profit more important than cost (ranking)
- weight must not exceed 2400g (constraint)



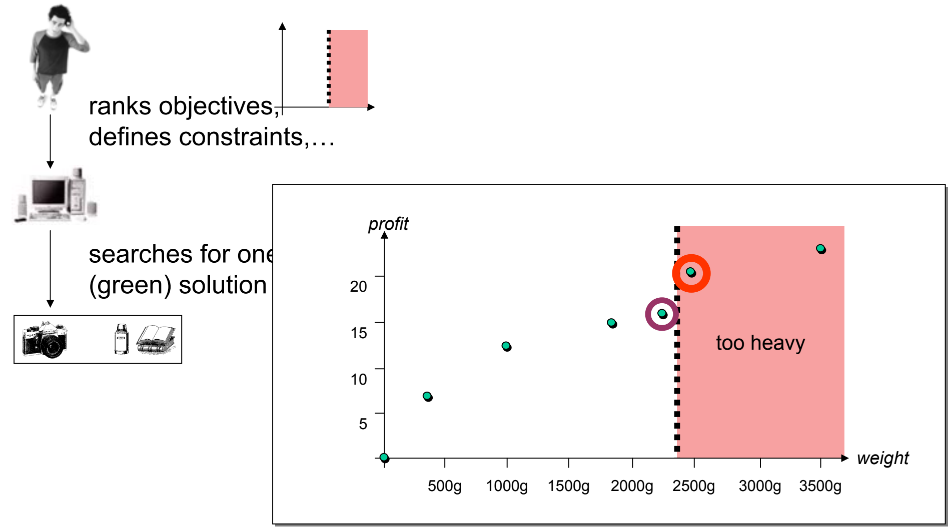
When to Make the Decision

Before Optimization:



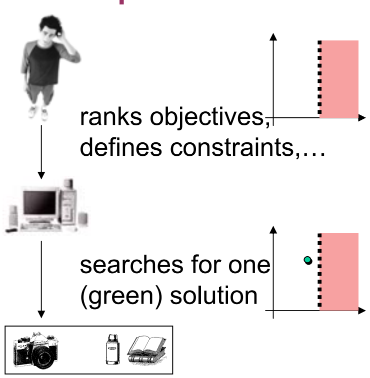
When to Make the Decision

Before Optimization:

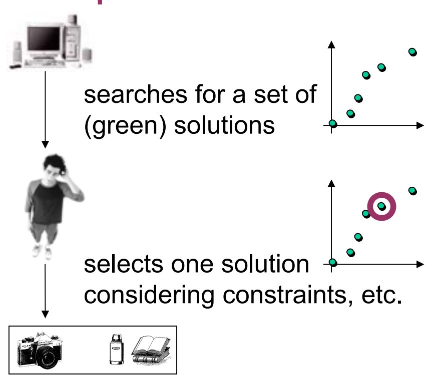


When to Make the Decision

Before Optimization:



After Optimization:



- decision making often easier
- EAs well suited

Outline

1. **Introduction:** Why multiple objectives make a difference
2. **Basic Principles:** Terms one needs to know
3. **Algorithm Design:** Do it yourself
4. **Performance Assessment:** How to compare algorithms
5. **Applications Domains:** Where EMO is useful
6. **Further Information:** What else

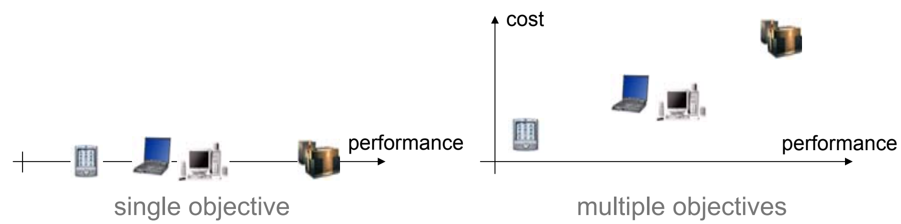
Optimization Problem: Definition

A general optimization problem is given by a quadruple *(X, Z, f, rel)* where

- *X* denotes the decision space containing the elements among which the best is sought; elements of *X* are called decision vectors or simply solutions;
- *Z* denotes the objective space, the space within which the decision vectors are evaluated and compared to each other; elements of *Z* are denoted as objective vectors;
- *f* represents a function $f: X \rightarrow Z$ that assigns each decision vector a corresponding objective vector; *f* is usually neither injective nor surjective;
- *rel* is a binary relation over *Z*, i.e., $rel \subseteq Z \times Z$, which represents a partial order over *Z*.

Objective Functions

- Usually, *f* consists of one or several functions $f_1, ..., f_n$ that assign each solution a real number. Such a function $f_i: X \rightarrow \Re$ is called an objective function, and examples are cost, size, execution time, etc.
- In the case of a single objective function (n=1), the problem is denoted as a single-objective optimization problem; a multiobjective optimization problem involves several (n ≥ 2) objective functions:

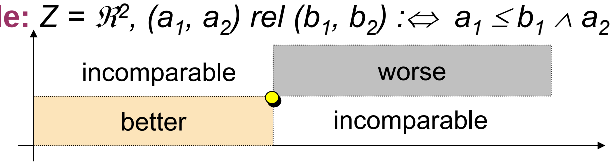


Comparing Objective Vectors

The pair *(Z, rel)* forms a partially ordered set, i.e., for any two objective vectors $a, b \in Z$ there can be four situations:

- *a* and *b* are equal: *a rel b* and *b rel a*
- *a* is better than *b*: *a rel b* and not *(b rel a)*
- *a* is worse than *b*: not *(a rel b)* and *b rel a*
- *a* and *b* are incomparable: neither *a rel b* nor *b rel a*

Example: $Z = \Re^2$, $(a_1, a_2)\ rel\ (b_1, b_2) :\Leftrightarrow a_1 \leq b_1 \wedge a_2 \leq b_2$



Often, *(Z, rel)* is a totally ordered set, i.e., for all $a, b \in Z$ either *a rel b* or *b rel a* or both holds (no incomparable elements).

Preference Structures

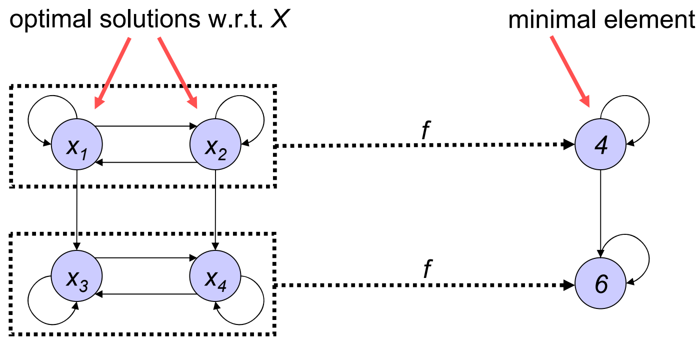
- The function *f* together with the partially ordered set *(Z, rel)* defines a preference structure on the decision space *X* that reflects which solutions the decision maker / user prefers to other solutions:

  $x_1\ prefrel\ x_2 :\Leftrightarrow f(x_1)\ rel\ f(x_2)$

- One says:
  - Two solutions x_1, x_2 are equal iff $x_1 = x_2$;
  - A solution x_1 is indifferent to a solution x_2 iff $x_1\ prefrel\ x_2$ and $x_2\ prefrel\ x_1$ and $x_1 \neq x_2$;
  - A solution x_1 is preferred to a solution x_2 iff $x_1\ prefrel\ x_2$;
  - A solution x_1 is strictly preferred to a solution x_2 iff $x_1\ prefrel\ x_2$ and not ($x_2\ prefrel\ x_1$);
  - A solution x_1 is incomparable to a solution x_2 iff neither $x_1\ prefrel\ x_2$ nor $x_2\ prefrel\ x_1$.

The Notion of Optimality

- A solution $x \in X$ is called optimal with respect to a set $S \subseteq X$ iff no solution $x' \in S$ is strictly preferred to x, i.e., for all $x' \in S$: x' *prefrel* $x \Rightarrow x$ *prefrel* x'.
- In other words, $f(x)$ is a minimal element of $f(S)$ regarding the partially ordered set (Z, rel).



Pareto Dominance

Assumption:

- n objective functions $f_i: X \rightarrow \Re$ where $Z = \Re^n$
- all objectives are to be maximized

Usually considered relation: weak Pareto dominance

- optimization problem: $(X, \Re^n, (f_1, \ldots, f_n), \preceq)$
- weak Pareto dominance:

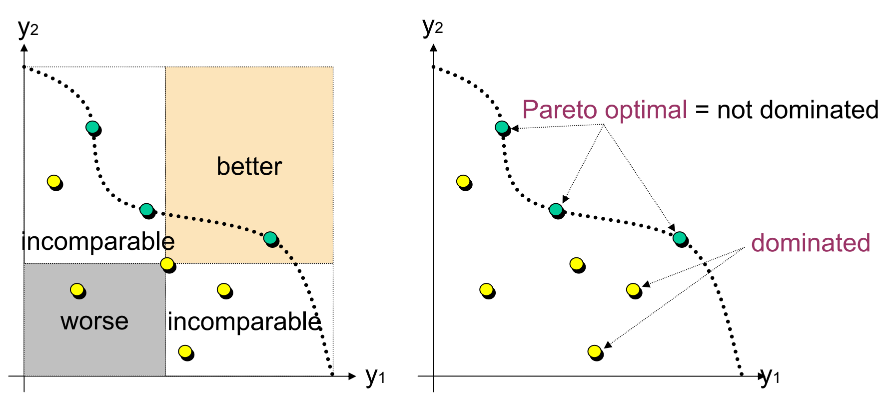
$$x_1 \preceq x_2 :\Leftrightarrow \forall 1 \leq i \leq n : f_i(x_1) \geq f_i(x_2)$$

- Pareto dominance: strict version of weak Pareto dominance

$$x_1 \prec x_2 :\Leftrightarrow x_1 \preceq x_2 \;\wedge\; x_2 \not\preceq x_1$$

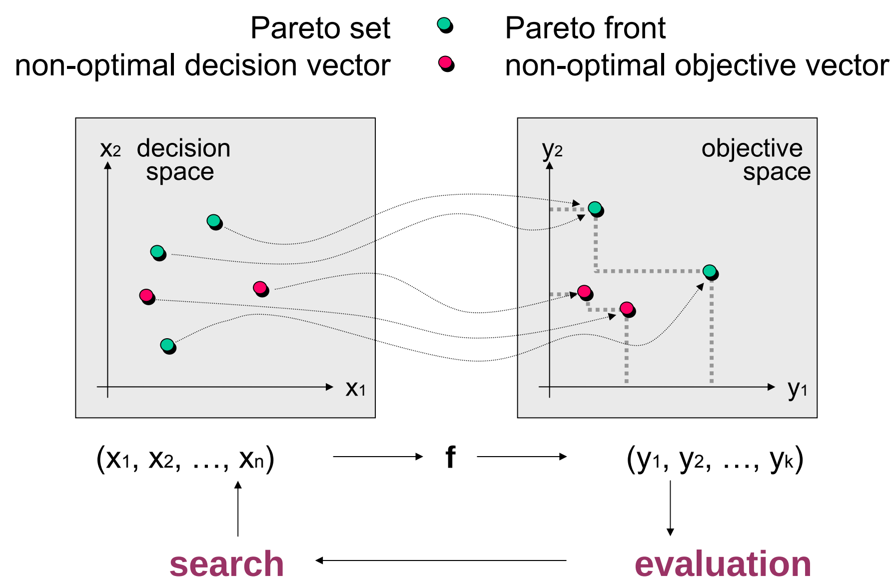
Illustration of Pareto Optimality

Maximize $(y_1, y_2, \ldots, y_k) = f(x_1, x_2, \ldots, x_n)$



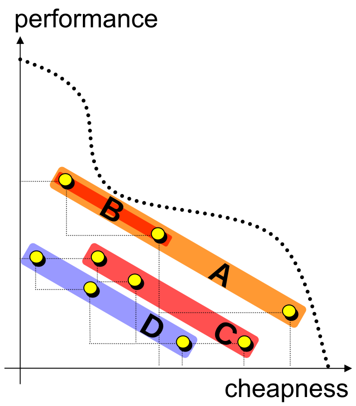
Pareto(-optimal) set = set of all Pareto-optimal solutions

Decision and Objective Space



Pareto Set Approximations

Pareto set approximation (algorithm outcome) =
set of incomparable solutions



A is **better** than **B**
= not worse in all objectives
and sets not equal

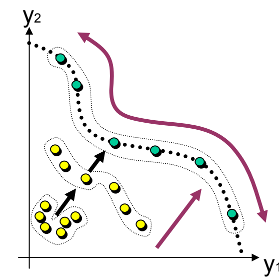
C **dominates** **D**
= better in at least one objective

A **strictly dominates** **C**
= better in all objectives

B is **incomparable** to **C**
= neither set weakly better

What Is the Optimization Goal?

- Find all Pareto-optimal solutions?
  - Impossible in continuous search spaces
  - How should the decision maker handle 10000 solutions?
- Find a representative subset of the Pareto set?
  - Many problems are NP-hard
  - What does representative actually mean?
- Find a good approximation of the Pareto set?
  - What is a good approximation?
  - How to formalize intuitive understanding:
    - ❶ close to the Pareto front
    - ❷ well distributed

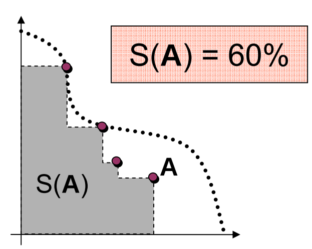


Preference Information

Preference information (here) = any additional information that refines the dominance relation on approximation sets (partial order → total order)

Example:

optimization goal
=
maximize size S of
dominated objective space

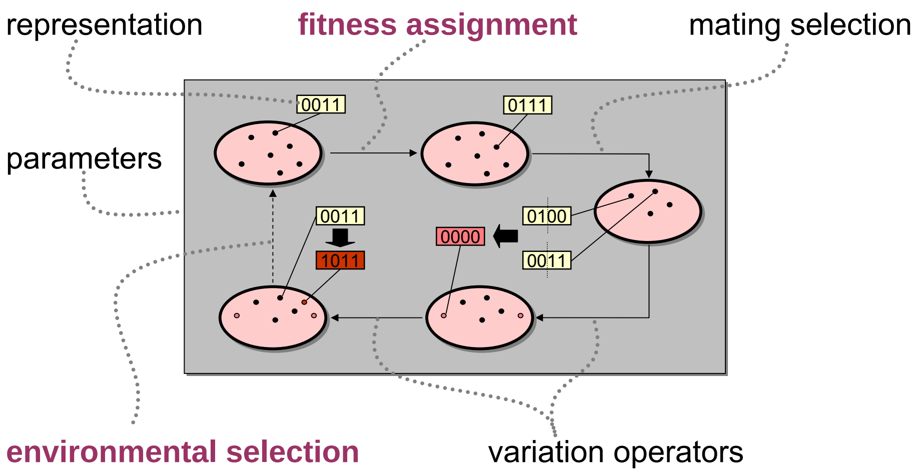


Note: every algorithm implicitly or explicitly makes assumptions about the decision maker's preferences (limited memory, selection)

Outline

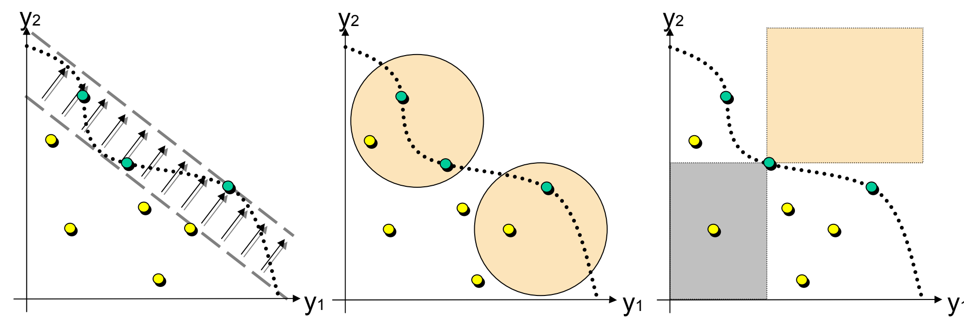
1. **Introduction:** Why multiple objectives make a difference
2. **Basic Principles:** Terms one needs to know
3. **Algorithm Design:** Do it yourself
4. **Performance Assessment:** How to compare algorithms
5. **Applications Domains:** Where EMO is useful
6. **Further Information:** What else

Design Choices



Ranking Solutions

aggregation-based *weighted sum* | **criterion-based** *VEGA* | **dominance-based** *SPEA2*



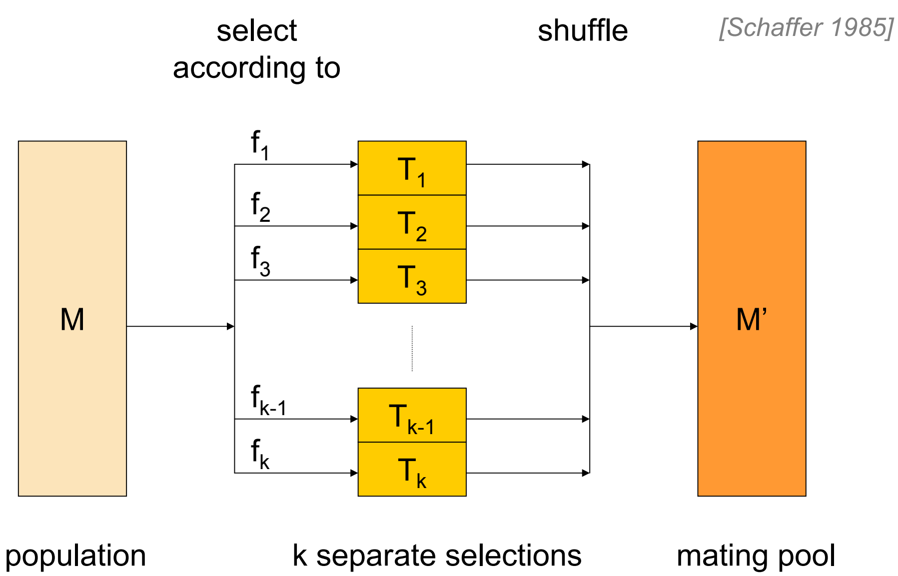
parameter-oriented
scaling-dependent

⟷

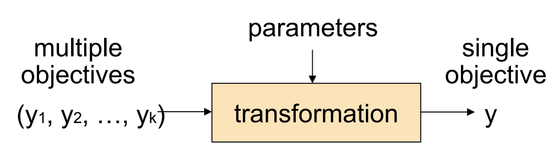
set-oriented
scaling-independent

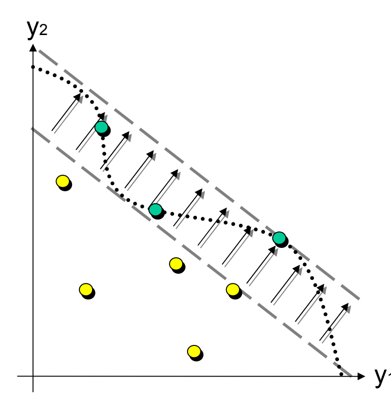
Example: VEGA

[Schaffer 1985]

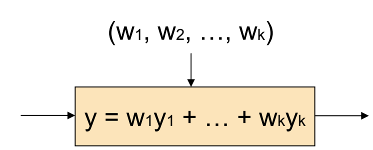


Aggregation-Based Ranking



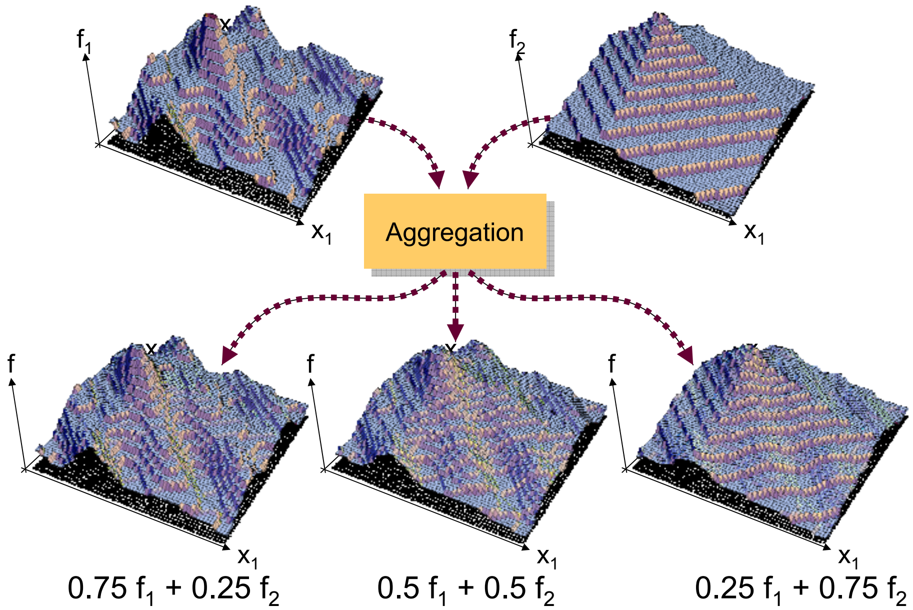


Example: weighting approach



Note: weights need to be varied during the run...

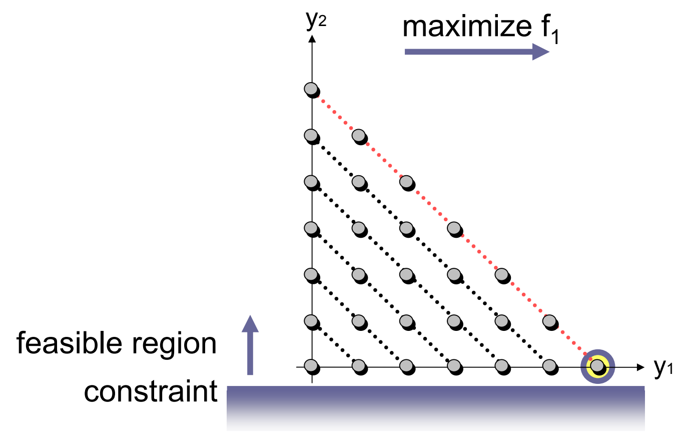
Example: Weighted Sum



Example: Multistart Constraint Method

Underlying concept:

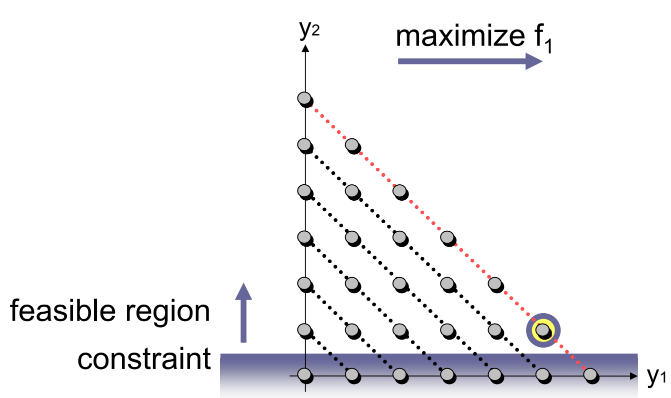
- Convert all objectives except of one into constraints
- Adaptively vary constraints



Example: Multistart Constraint Method

Underlying concept:

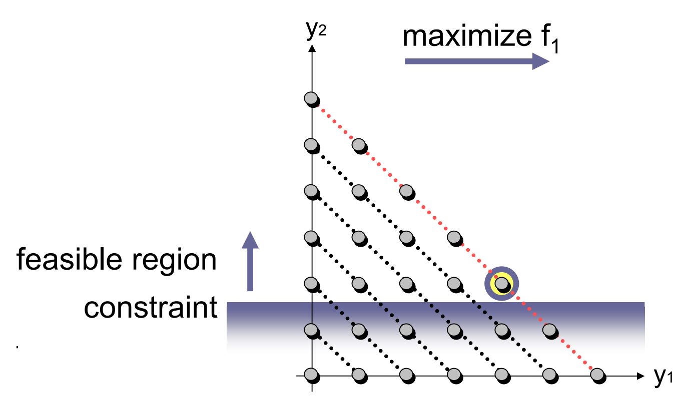
- Convert all objectives except of one into constraints
- Adaptively vary constraints



Example: Multistart Constraint Method

Underlying concept:

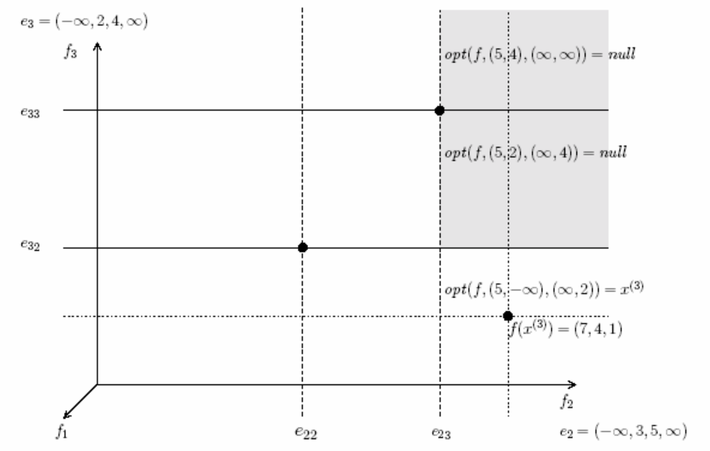
- Convert all objectives except of one into constraints
- Adaptively vary constraints



Example: Multistart Constraint Method (Cont'd)

Extension to n objectives: ECEA *[Laumanns et al. 2006]*

- f_1 is the objective to optimize
- The boxes are defined by constraints on f_2 and f_3



Dominance-Based Ranking

Types of information:

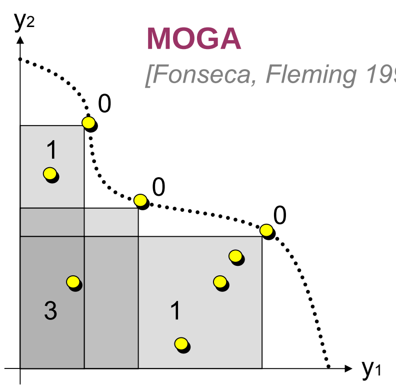
- **dominance rank** — by how many individuals is an individual dominated?
- **dominance count** — how many individuals does an individual dominate?
- **dominance depth** — at which front is an individual located?

Examples:

- *MOGA, NPGA* — dominance rank
- *NSGA/NSGA-II* — dominance depth
- *SPEA/SPEA2* — dominance count + rank

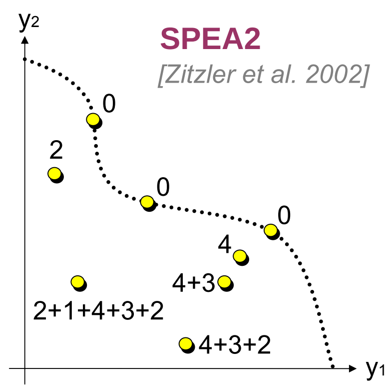
Example: MOGA and SPEA2

MOGA
[Fonseca, Fleming 1993]



R (raw fitness) =
#dominating solutions

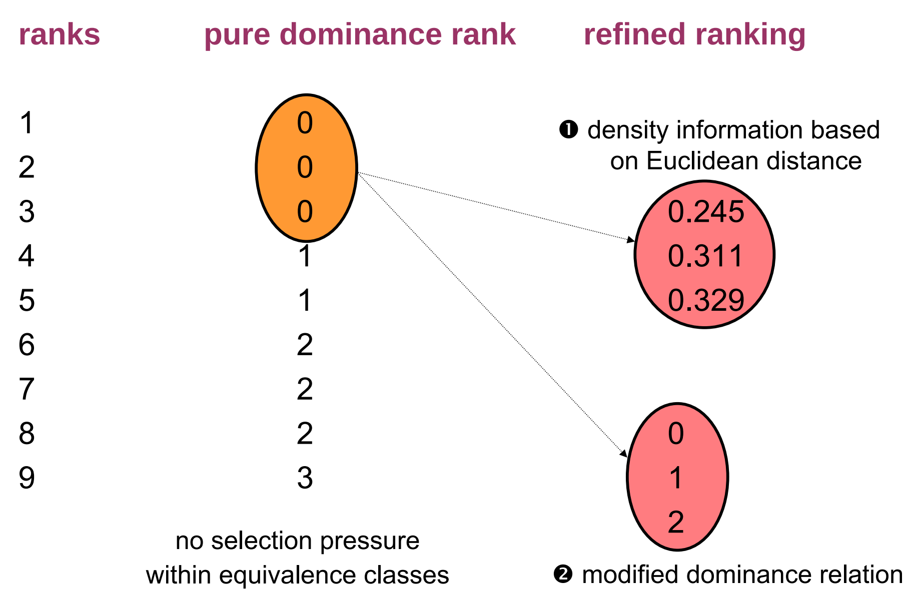
SPEA2
[Zitzler et al. 2002]



S (strength) =
#dominated solutions

R (raw fitness) =
$\sum$ strengths of dominators

Refining Rankings

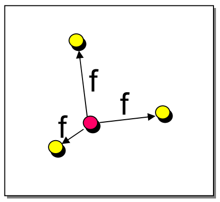


Methods Based On Euclidean Distance

Density estimation techniques: *[Silverman 1986]*

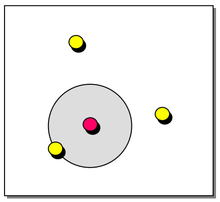
Kernel
MOGA

density estimate = sum of f values where f is a function of the distance



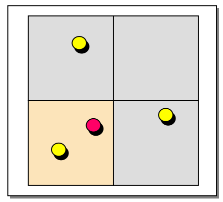
Nearest neighbor
SPEA2

density estimate = volume of the sphere defined by the nearest neighbor



Histogram
PAES

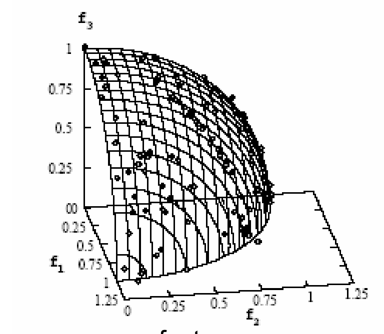
density estimate = number of solutions in the same box



Computation Effort Versus Accuracy

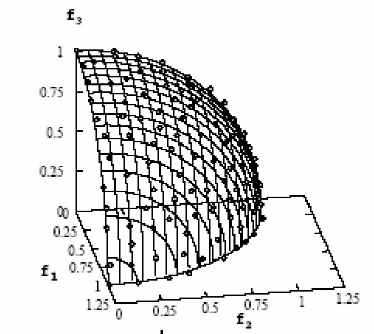
Two Nearest Neighbor Variants

Objective-Wise
NSGA-II



faster
good for 2 objectives
[Deb et al. 2002]

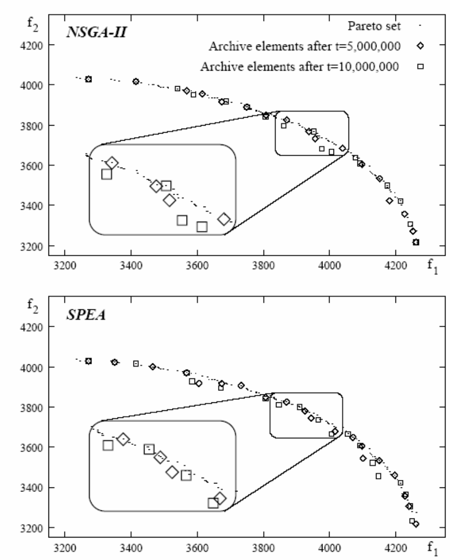
Euclidean Distance
SPEA2



slower
good for 3 objectives and more
[Zitzler et al. 2002]

The Problem of Deterioration

Observation:
The use of Euclidean distance can lead to deterioration



Knapsack problem
[Laumanns et al. 2002]

Refinement of Dominance Relations

Integration of Goals, Priorities, Constraints:
[Fonseca, Fleming 1998]

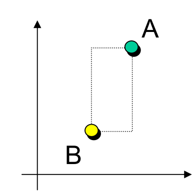
A is preferable over B $\Leftrightarrow (\overset{\boldsymbol{u}}{\frown}{\boldsymbol{u}_p}\ _p < \overset{\boldsymbol{u}}{\frown}{\boldsymbol{v}_p}) \vee \{(\overset{\boldsymbol{u}}{\frown}{\boldsymbol{u}_p} = \overset{\boldsymbol{u}}{\frown}{\boldsymbol{v}_p})$
$\wedge [(\overset{\boldsymbol{u}}{\smile}{\boldsymbol{v}_p} \not\leq \overset{\boldsymbol{u}}{\smile}{\boldsymbol{g}_p}) \vee (\boldsymbol{u}_{1,\cdots,p-1} \underset{\boldsymbol{g}_{1,\cdots,p-1}}{\prec} \boldsymbol{v}_{1,\cdots,p-1})]\}$

Continuous dominance "relations": *[Zitzler et al. 2003]*

$I_{\varepsilon+}(A,B) = \min_i f_i(A) - f_i(B)$

$I_{\varepsilon+}(A,B) \geq 0$ and $I_{\varepsilon+}(B,A) < 0 \Leftrightarrow$ A dominates B

(binary additive epsilon quality indicator)



Example: IBEA

Question: How to continuous dominance "relations" for fitness assignment? *[Zitzler, Künzli 2004]*

Given: function I (binary quality indicator) with

A dominates B ⇔ I(A, B) < I(B, A)

Idea: measure for "loss in quality" if A is removed

Fitness: $F'(\boldsymbol{x}^1) = \sum_{\boldsymbol{x}^2 \in P \setminus \{\boldsymbol{x}^1\}} I(\{\boldsymbol{x}^2\}, \{\boldsymbol{x}^1\})$

...corresponds to continuous extension of dominance rank

...blurrs influence of dominating and dominated individuals

Example: IBEA (Cont'd)

Fitness assignment: $O(n^2)$

Fitness: $$F(\boldsymbol{x}^1) = \sum_{\boldsymbol{x}^2 \in P \setminus \{\boldsymbol{x}^1\}} -e^{-I(\{\boldsymbol{x}^2\}, \{\boldsymbol{x}^1\})/\kappa}$$

- parameter κ is problem- and indicator-dependent
- no additional diversity preservation mechanism

Mating selection: O(n)

- binary tournament selection, fitness values constant

Environmental selection: $O(n^2)$

- iteratively remove individual with lowest fitness
- update fitness values of remaining individuals after each deletion

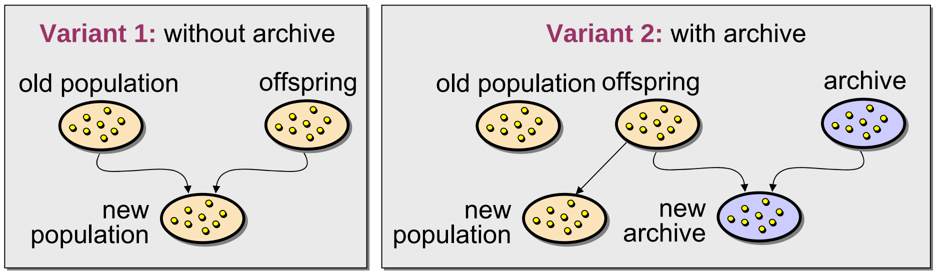
Further Design Aspects

- **Constraint handling:**
  How to integrate constraints into fitness assignment?
- **Archiving / environmental selection:**
  How to keep a good approximation?
- **Hybridization:**
  How to integrate, e.g., local search in a multiobjective EA?
- **Preference articulation:**
  How to focus the search on interesting regions?
- **Robustness and uncertainty:**
  How to account for variations in the objective function values?
- **Data structures:**
  How to support, e.g., fast dominance checks?

Constraint Handling & Multiple Objectives

	penalty functions Add penalty term to fitness	**constraints as objectives** Introduce additional objective(s)	**modified dominance** extend to infeasible solutions
overall constraint violation	*[Michalewicz 1992]*	*[Wright, Loosemore 2001]*	*[Deb 2001]*
constraints treated separately	?	*[Coello 2000]*	*[Fonseca, Fleming 1998]*

Archiving / Environmental Selection



deterministic truncation

archive = only nondominated solutions

Additional selection criteria:

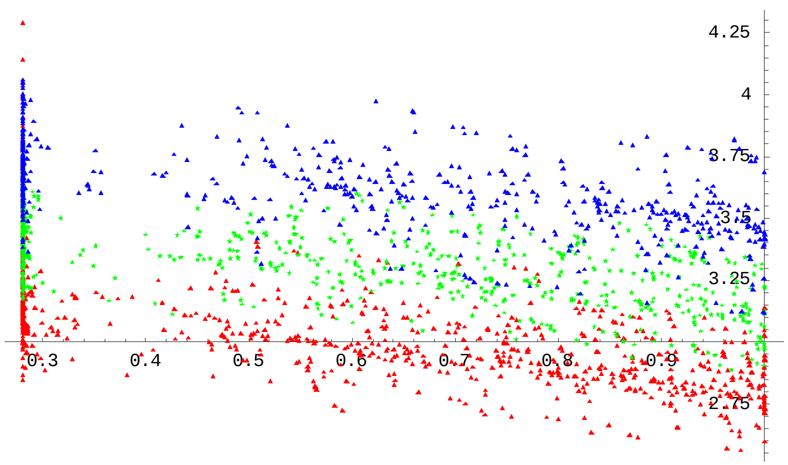
- Density information / other preferences
- Time
- Chance

Outline

1. **Introduction:** Why multiple objectives make a difference
2. **Basic Principles:** Terms one needs to know
3. **Algorithm Design:** Do it yourself
4. **Performance Assessment:** How to compare algorithms
5. **Applications Domains:** Where EMO is useful
6. **Further Information:** What else

Once Upon a Time...

... multiobjective EAs were mainly compared visually:



ZDT6 benchmark problem: IBEA, SPEA2, NSGA-II

Performance Assessment: Approaches

❶ **Theoretically (by analysis):** difficult

- Limit behavior (unlimited run-time resources)
- Running time analysis

❷ **Empirically (by simulation):** standard

Problems: randomness, multiple objectives

Issues: quality measures, statistical testing, visualization, benchmark problems, parameter settings, …

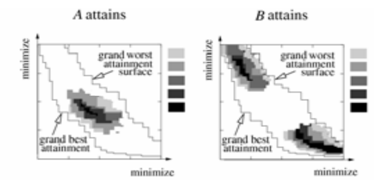
Two Approaches for Empirical Studies

Attainment function approach:

- Applies statistical tests directly to the samples of approximation sets
- Gives detailed information about how and where performance differences occur

Quality indicator approach:

- First, reduces each approximation set to a single value of quality
- Applies statistical tests to the samples of quality values

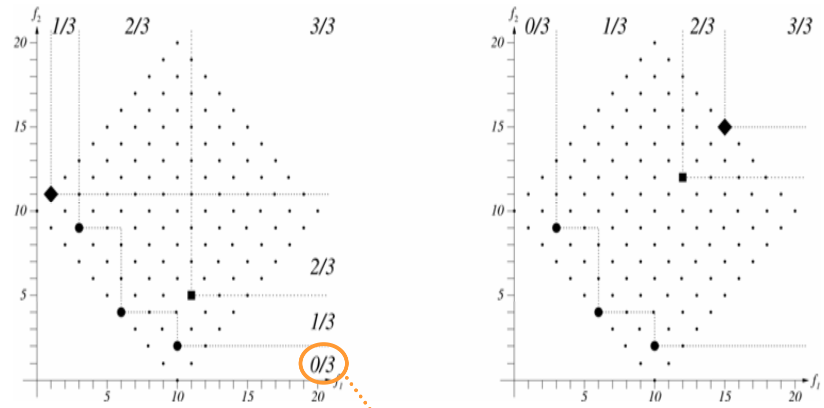


Indicator	A	B
Hypervolume indicator	6.3431	7.1924
ϵ-indicator	1.2090	0.12722
R_2 indicator	0.2434	0.1643
R_3 indicator	0.6454	0.3475

© Eckart Zitzler
ETH Zürich
GECCO 2006
Tutorial on EMO

Empirical Attainment Functions

three runs of two multiobjective optimizers

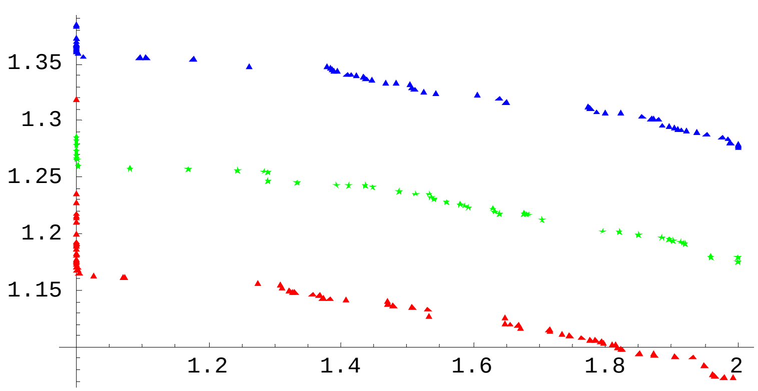


frequency of attaining regions

© Eckart Zitzler
ETH Zürich
GECCO 2006
Tutorial on EMO

Attainment Plots

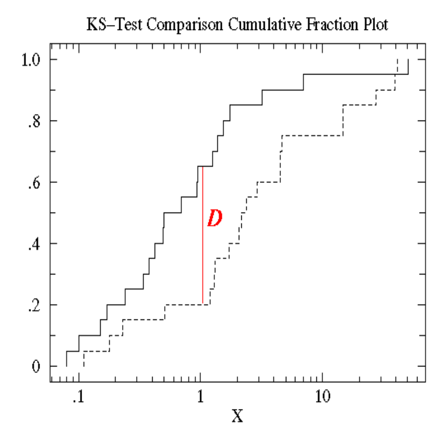
50% attainment surface for IBEA, SPEA2, NSGA2 (ZDT6)



© Eckart Zitzler
ETH Zürich
GECCO 2006
Tutorial on EMO

Attainment Function Analysis

- A Kolmogorov-Smirnov test examines the maximum difference between two cumulative distribution functions
- A KS-like test can be used to probe differences between the empirical attainment functions of a pair of optimizers, *A* and *B*
- The null hypothesis is that the attainment functions of *A* and *B* are identical
- The alternative hypothesis is that the distributions differ somewhere



[Fonseca et al. 2001]

Statistical Assessment

ZDT6

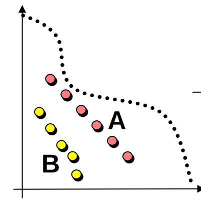
- IBEA – NSGA-II
  - significant difference (p=0)
- IBEA – SPEA2
  - significant difference (p=0)
- SPEA2 – NSGA-II
  - significant difference (p=0)

Knapsack

- IBEA – NSGA-II
  - no significant difference
- IBEA – SPEA2
  - no significant difference
- SPEA2 – NSGA-II
  - no significant difference

Quality Indicator Approach

Goal: compare two Pareto set approximations A and B



→

	A	B
hypervolume	*432.34*	*420.13*
distance	*0.3308*	*0.4532*
diversity	*0.3637*	*0.3463*
spread	*0.3622*	*0.3601*
cardinality	*6*	*5*

→ "**A** better"

Comparison method C = quality measure(s) + Boolean function

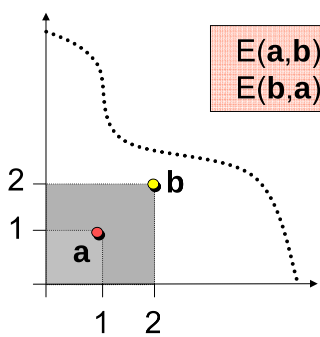
A, B —quality measure→ R^n —Boolean function→ statement

reduction **interpretation**

Example: ε-Quality Indicator

Two solutions:

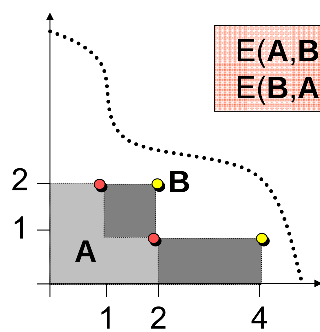
E(**a**,**b**) =
$\max_{1 \leq i \leq n} \min_{\varepsilon} \varepsilon \cdot f_i(\mathbf{a}) \geq f_i(\mathbf{b})$



E(**a**,**b**) = 2
E(**b**,**a**) = ½

Two approximations:

E(**A**,**B**) =
$\max_{\mathbf{b} \in \mathbf{B}} \min_{\mathbf{a} \in \mathbf{A}} E(\mathbf{a},\mathbf{b})$



E(**A**,**B**) = 2
E(**B**,**A**) = ½

Unary quality indicator: I(A) = E(A,R) where R is a reference set

[Zitzler et al. 2003]

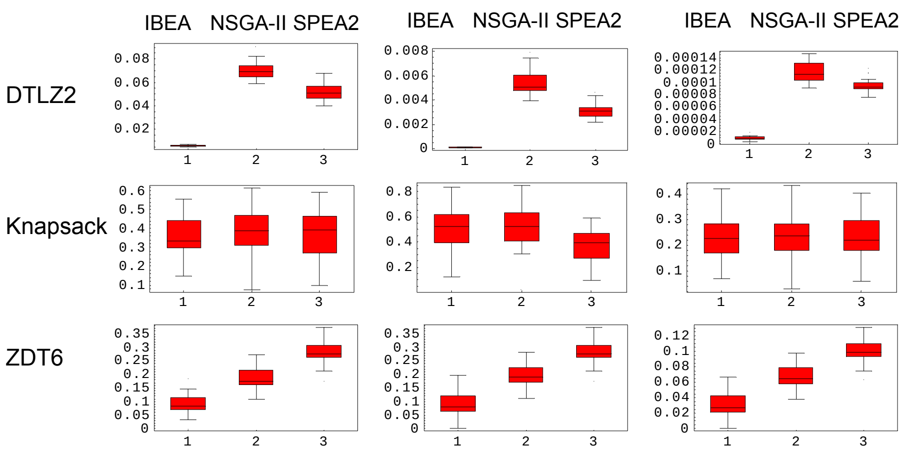
Power of Unary Quality Indicators

Important: compliance with dominance relations *[Zitzler et al. 2003]*

indicator	name / reference	Boolean function	compatibility	completeness
I_{HC}	enclosing hypercube indicator / Section III-B.1	$I_2^{HC}(A) < I_1^{HC}(B)$	$\succ\succ$	-
I_O	objective vector indicator / Section III-B.1	$I_1^{O}(A) < I_1^{O}(B)$	$\succ\succ$	-
I_H	hypervolume indicator / [7]	$I_H(A) > I_H(B)$	$\not\triangleright$	$\triangleright$
I_W	average best weight combination / [19]	$I_W(A) < I_W(B)$	$\not\triangleright$	$\succ\succ$
I_D	distance from reference set / [20]	$I_D(A) < I_D(B)$	$\not\triangleright$	$\succ\succ$
$I_{\epsilon 1}$	unary ϵ-indicator / Section III-B.2	$I_{\epsilon 1}(A) < I_{\epsilon 1}(B)$	$\not\triangleright$	$\succ\succ$
I_{PF}	fraction of Pareto-optimal front covered / [22]	$I_{PF}(A) > I_{PF}(B)$	$\not\triangleright$	-
I_P	number of Pareto points contained / Section III-B.2	$I_P(A) > I_P(B)$	$\not\triangleright$	-
I_{ER}	error ratio / [13]	$I_{ER}(A) > 0$	$\not\succ$	-
I_{CD}	chi-square-like deviation indicator / [14]	$I_{CD}(A) < I_{CD}(B)$	-	-
I_S	spacing / [23]	$I_S(A) < I_S(B)$	-	-
I_{ONVG}	overall nondominated vector generation / [13]	$I_{ONVG}(A) > I_{ONVG}(B)$	-	-
I_{GD}	generational distance / [13]	$I_{GD}(A) < I_{GD}(B)$	-	-
I_{ME}	maximum Pareto front error / [13]	$I_{ME}(A) < I_{ME}(B)$	-	-
I_{MS}	maximum spread / [21]	$I_{MS}(A) > I_{MS}(B)$	-	-
I_{MD}	minimum distance between two solutions / [24]	$I_{MD}(A) > I_{MD}(B)$	-	-
I_{CE}	coverage error / [24]	$I_{CE}(A) < I_{CE}(B)$	-	-
I_{DU}	deviation from uniform distribution / [25]	$I_{DU}(A) < I_{DU}(B)$	-	-
I_{OS}	Pareto spread / [26]	$I_{OS}(A) > I_{OS}(B)$	-	-
I_A	accuracy / [26]	$I_A(A) > I_A(B)$	-	-
I_{NDC}	number of distinct choices / [26]	$I_{NDC}(A) > I_{NDC}(B)$	-	-
I_{CL}	cluster / [26]	$I_{CL}(A) < I_{CL}(B)$	-	-

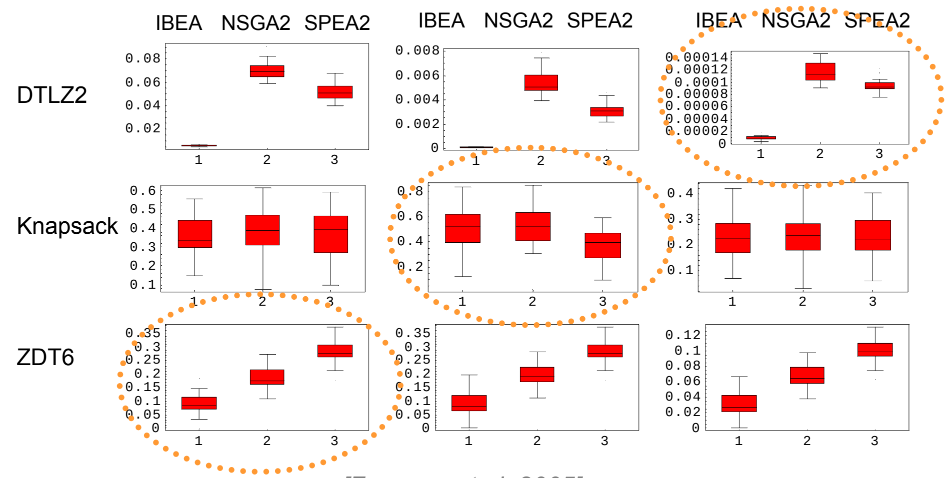
strictly better · not weakly better · not better · weakly better

Example: Box Plots



[Fonseca et al. 2005]

Example: Box Plots



[Fonseca et al. 2005]

Statistical Assessment (Kruskal Test)

ZDT6
Epsilon

is better than

	IBEA	NSGA2	SPEA2
IBEA		~0	~0
NSGA2	1		~0
SPEA2	1	1	

Overall p-value = 6.22079e-17.
Null hypothesis rejected (alpha 0.05)

DTLZ2
R

is better than

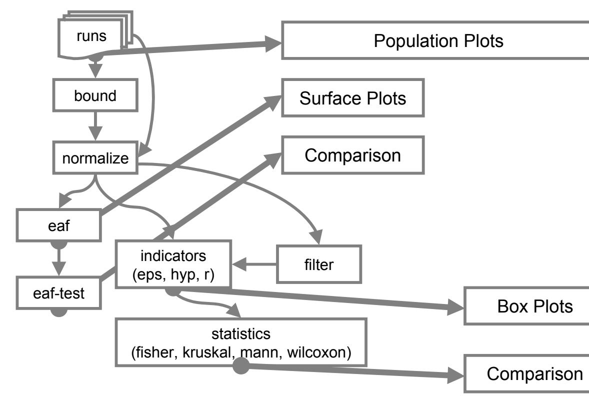
	IBEA	NSGA2	SPEA2
IBEA		~0	~0
NSGA2	1		1
SPEA2	1	~0	

Overall p-value = 7.86834e-17.
Null hypothesis rejected (alpha 0.05)

Knapsack/Hypervolume: H0 = No significance of any differences

Performance Assessment Tools

- Reference set calculation
- Attainment function calculation
- Indicators
- Statistical testing procedures

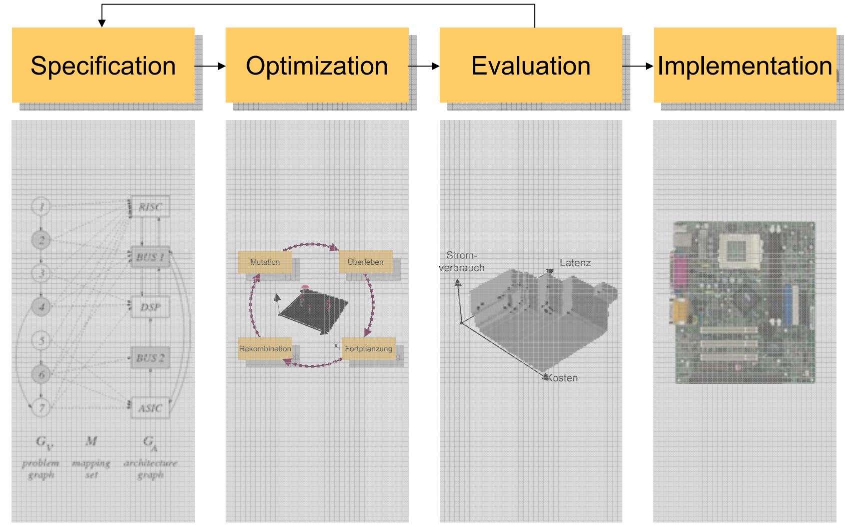


http://www.tik.ee.ethz.ch/pisa

Outline

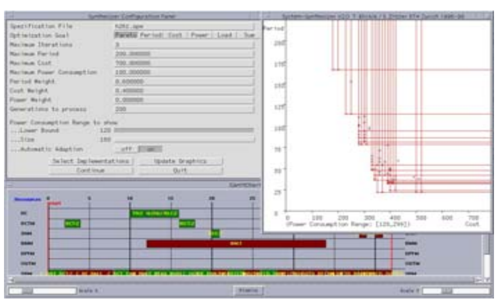
1. **Introduction:** Why multiple objectives make a difference
2. **Basic Principles:** Terms one needs to know
3. **Algorithm Design:** Do it yourself
4. **Performance Assessment:** How to compare algorithms
5. **Applications Domains:** Where EMO is useful
6. **Further Information:** What else

Design Space Exploration



Examples: computer design, biological experiment design, etc.

Example Applications

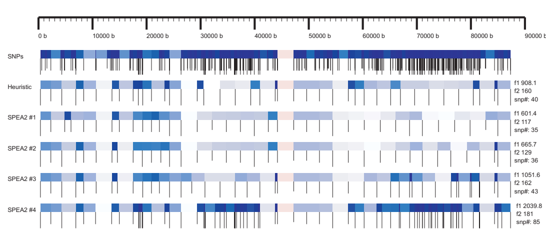


Architecture exploration:
- min. cost
- max. performance
- min. power consumption

[Eisenring, Thiele, Zitzler 2000]

Genetic marker selection:
- min. cost
- max. sensitivity

[Hubley, Zitzler, Roach 2003]

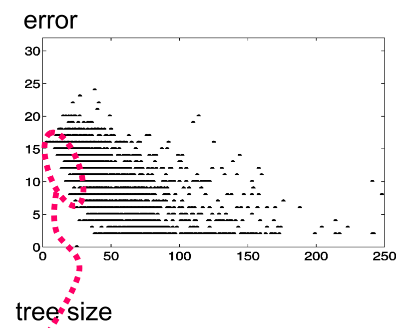


Application: Genetic Programming

Problem: Trees grow rapidly
- Premature convergence
- Overfitting of training data

Common approaches:
- Constraint
  (tree size limitation)
- Penalty term
  (parsimony pressure)
- Objective ranking
  (size post-optimization)
- Structure-based (ADF, etc.)

Multiobjective approach:
Optimize both error and size



Keep and optimize small trees
(potential building blocks)

Multiobjective GP: Results

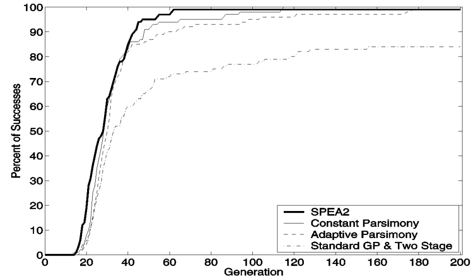
Multiobjective approach (SPEA2) can find

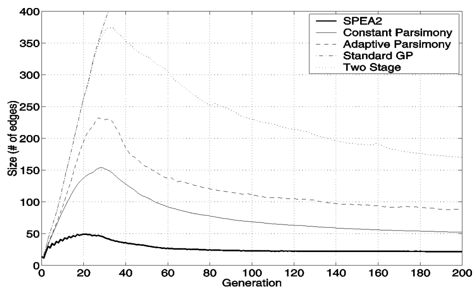
- a correct solution with higher probability
- a correct solution slightly faster

- more compact (correct) solutions

than alternative approaches on even-parity problem.

[Bleuler et al. 2001]

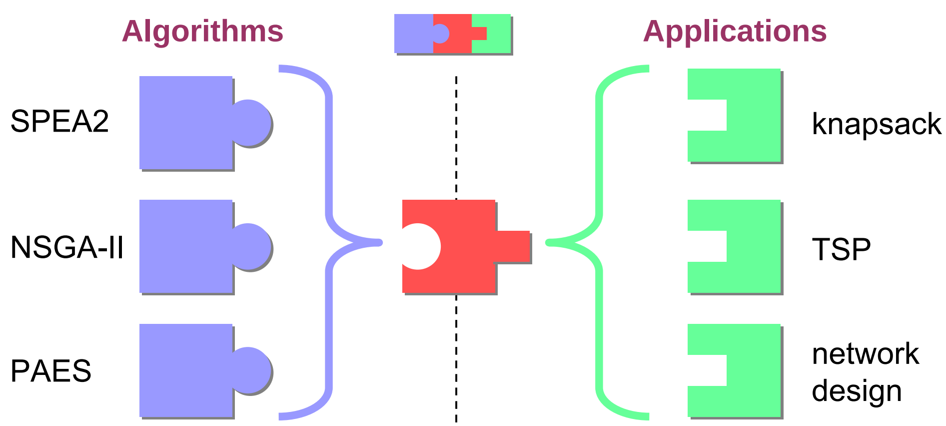




Outline

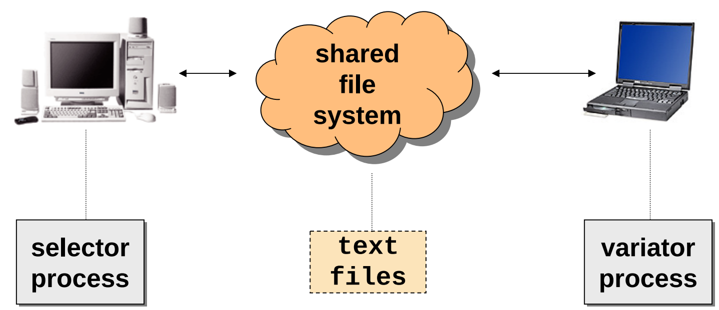
1. **Introduction:** Why multiple objectives make a difference
2. **Basic Principles:** Terms you need to know
3. **Algorithm Design:** Do it yourself
4. **Performance Assessment:** Once upon a time
5. **Applications Domains:** Where EMO is useful
6. **Further Information:** What else

The Concept of PISA



text-based
Platform and programming language independent **I**nterface for **S**earch **A**lgorithms *[Bleuler et al.: 2003]*

PISA: Implementation

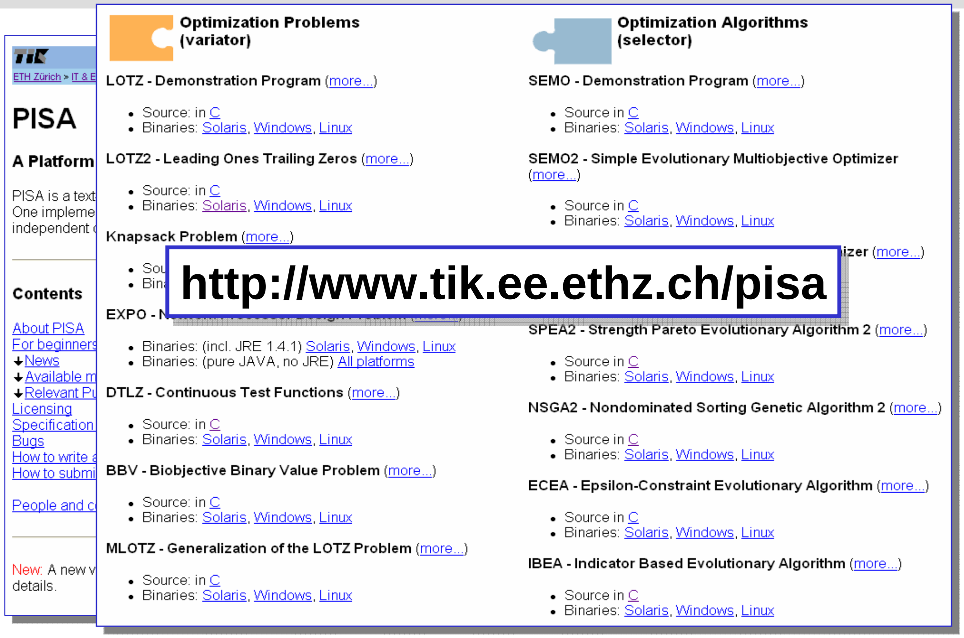


application independent:
- mating / environmental selection
- individuals are described by IDs and objective vectors

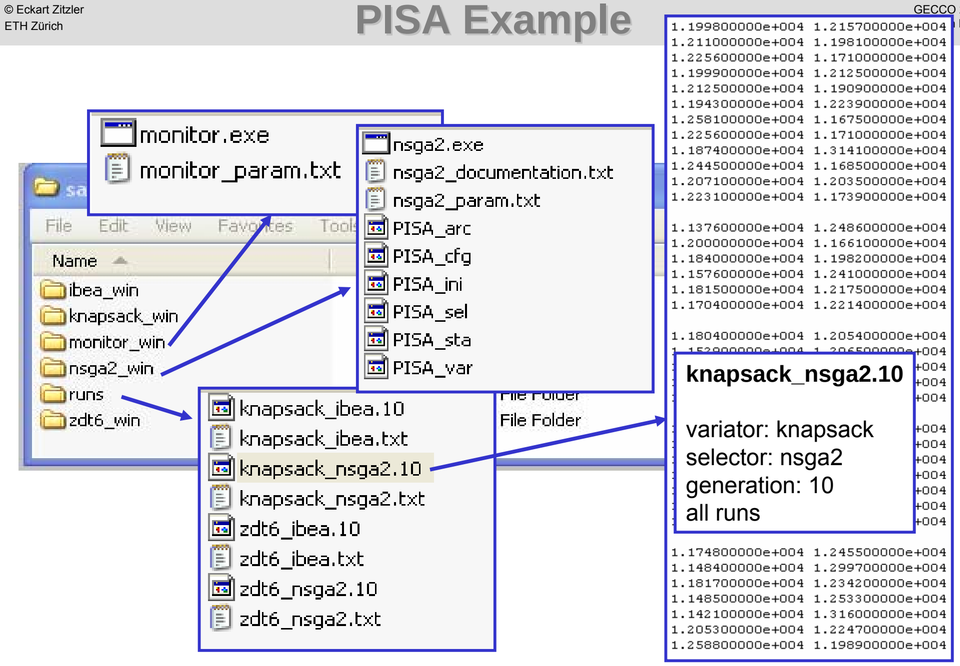
handshake protocol:
- state / action
- individual IDs
- objective vectors
- parameters

application dependent:
- variation operators
- stores and manages individuals

PISA Website



PISA Example



The EMO Community

Links:

- EMO mailing list:
  http://w3.ualg.pt/lists/emo-list/
- EMO bibliography:
  http://www.lania.mx/~ccoello/EMOO/

Events:

- Conference on Evolutionary Multi-Criterion Optimization (EMO 2007 to be held in Japan)

Books:

- ***Multi-Objective Optimization using Evolutionary Algorithms***
  Kalyanmoy Deb, Wiley, 2001
- ***Evolutionary Algorithms for Solving Multi Evolutionary Algorithms for Solving Multi-Objective Problems Objective Problems***, Carlos A. Coello Coello, David A. Van Veldhuizen & Gary B. Lamont, Kluwer, 2002

References

- Bleuler, S., Brack, M.,Thiele, L., Zitzler, E. (2001). Multiobjective Genetic Programming: Reducing Bloat Using SPEA2. CEC-2001, pp. 536 - 543.
- Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA - A Platform and Programming Language Independent Interface for Search Algorithms. EMO 2003, pp. 494 - 508.
- Coello, C. (2000). Treating Constraints as Objectives for Single-Objective Evolutionary Optimization, Engineering Optimization, 32(3):275-308.
- Deb, K. (2001). Multi -Objective Optimization using Evolutionary Algorithms.Wiley.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Volume 6, Issue 2, pp.182 – 197.
- Eisenring, M., Thiele, L., Zitzler, E. (2000). Handling Conflicting Criteria in Embedded System Design. IEEE Design & Test of Computers, Vol. 17, No. 2, pp. 51-59.
- Fonseca, C. M., Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest (Ed.), ICGA Proceedings, pp. 416–423.
- Fonseca, C. M., Fleming, P. J. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms—part i: A unified formulation. IEEE Transactions on Systems, Man, and Cybernetics 28(1), pp. 26–37.
- Fonseca, C., Knowles, J., Thiele, L., Zitzler, E. (2005). A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. EMO 2005.
- Grunert da Fonseca, V., Fonseca, C., Hall, A. (2001). Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. EMO 2001, pp. 213-225.
- Hubley, R., Zitzler, E., Roach, J. (2003). Evolutionary algorithms for the selection of single nucleotide polymorphisms. BMC Bioinformatics, Vol. 4, No. 30.
- Laumanns, M., Thiele, L., Deb, K., Zitzler, E. (2002). Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation, Vol. 10, No. 3, pp. 263—282.
- Laumanns, M., Thiele, L., Zitzler, E. (2006). An Efficient Metaheuristic Based on the Epsilon-Constraint Method. European Journal of Operational Research, Volume 169, Issue 3 , pp 932-942.
- Michalewicz, Z. (1992). Genetic Algorithms + Data Structures = Evolution Programs. Springer.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette (Ed.), ICGA Proceedings, pp. 93–100.
- Silverman, B. W. (1986). Density estimation for statistics and data analysis, Chapman and Hall, London.
- Wright, J., Loosemore, H. (2001). An Infeasibility Objective for Use in Constrained Pareto Optimization. EMO 2001, pp. 256-268.
- Zitzler, E., Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. PPSN VIII, pp. 832-842.
- Zitzler, E., Laumanns, M., Thiele, L. (2002). SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. Evolutionary Methods for Design, Optimisation, and Control, CIMNE, Barcelona, Spain, pages 95-100.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation, Vol. 7, No. 2, pages 117-132.