A Tutorial on Evolutionary Multiobjective Optimization

Eckart Zitzler

Computer Engineering and Networks Lab
Swiss Federal Institute of Technology (ETH) Zurich

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

TIK
Computer Engineering
and Networks Laboratory

Overview

❶ **The Big Picture:**

Optimization and evolutionary computation

❷ **The Construction Kit:**

Design issues and algorithmic concepts

❸ **The Pieces Put Together:**

Example of an algorithm variant

❹ **The Big Question:**

Performance of evolutionary algorithms

❺ **The Challenge:**

Standard interface for search algorithms

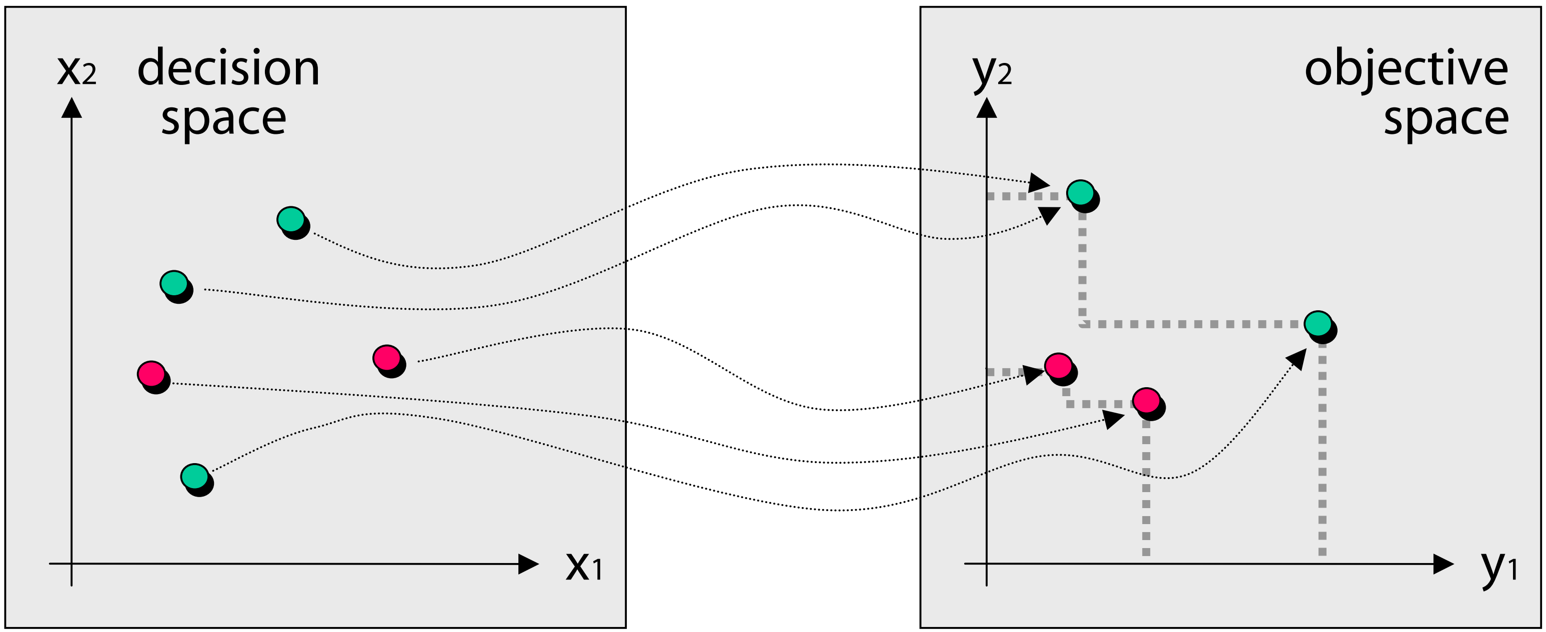
❻ **The End:**

Conclusions and outlook

Decision and Objective Space

Pareto set — Pareto front

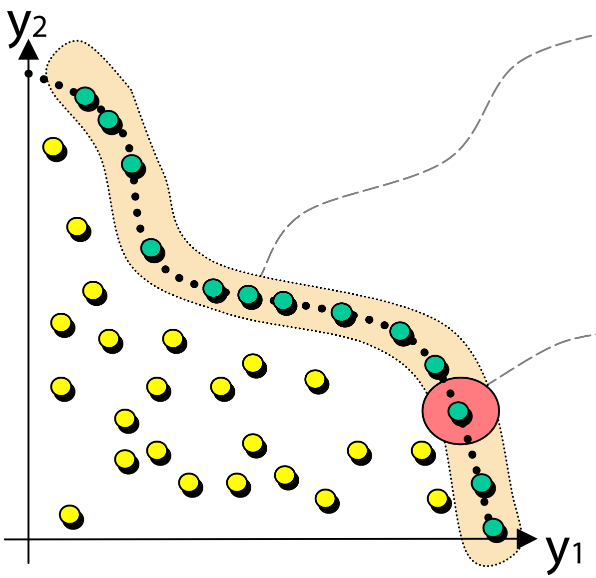
Pareto set approximation — Pareto front approximation



$(x_1, x_2, \ldots, x_n) \longrightarrow \mathbf{f} \longrightarrow (y_1, y_2, \ldots, y_k)$

search ⟵ **evaluation**

Optimization and Decision Making



Pareto optimality:
defines set of optimal trade-offs
(all objectives equally important)

Decision making:
choose best compromise
(based on preference information)

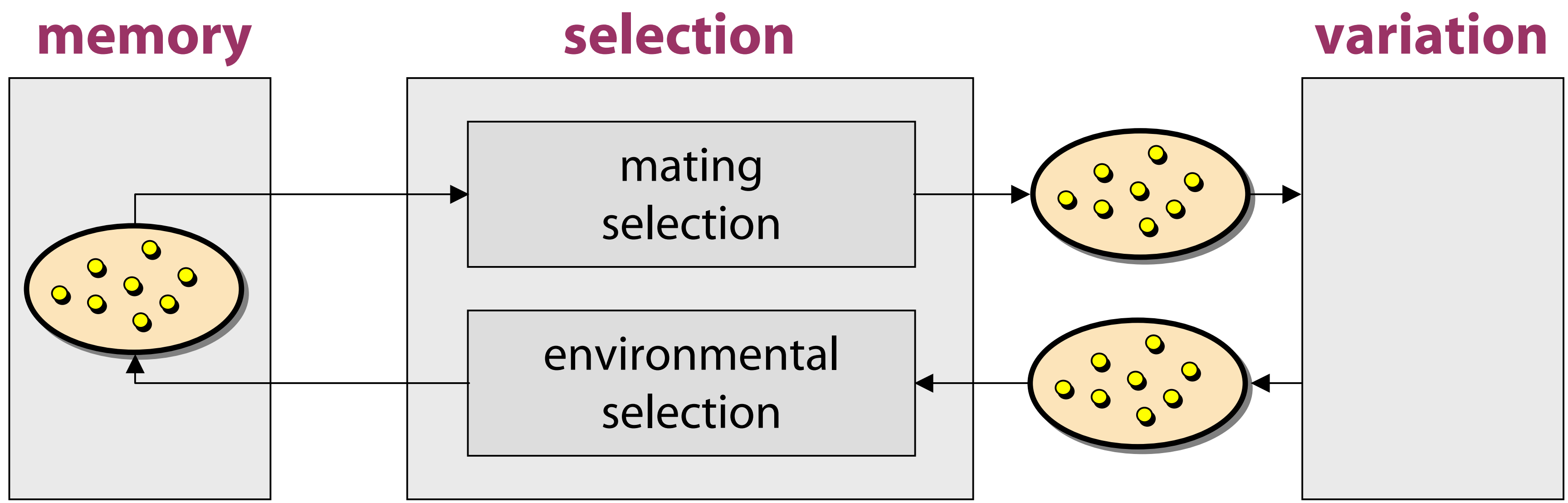
❶ Decision making before search (define single objective)

❷ Decision making after search (find/approximate Pareto set first)

❸ Decision making during search (guide search interactively)

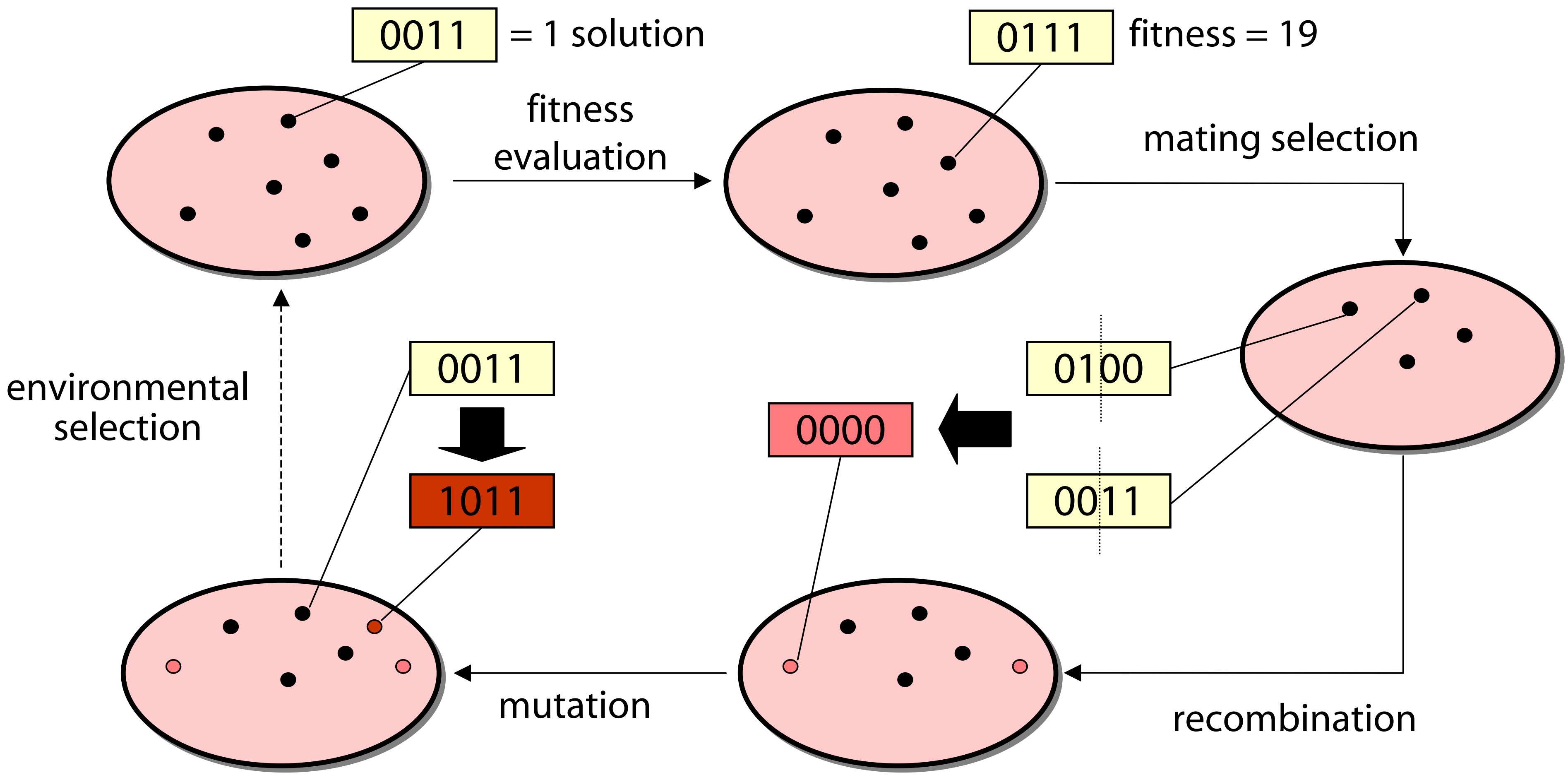
❹ Combinations of the above

Stochastic Search Algorithms



EA	≥ 1	both	≥ 1 ≥ 1	N : M randomized
TS	1	no mating selection	1 ≥ 1	1 : M deterministic
SA	1	no mating selection	1 ≥ 1	1 : M randomized
ACO	1	neither	1 1	1 : 1 randomized

Outline of a Simple Evolutionary Algorithm



Overview

❶ **The Big Picture:**

Optimization and evolutionary computation

❷ **The Construction Kit:**

Design issues and algorithmic concepts

❸ **The Pieces Put Together:**

Example of an algorithm variant

❹ **The Big Question:**

Performance of evolutionary algorithms

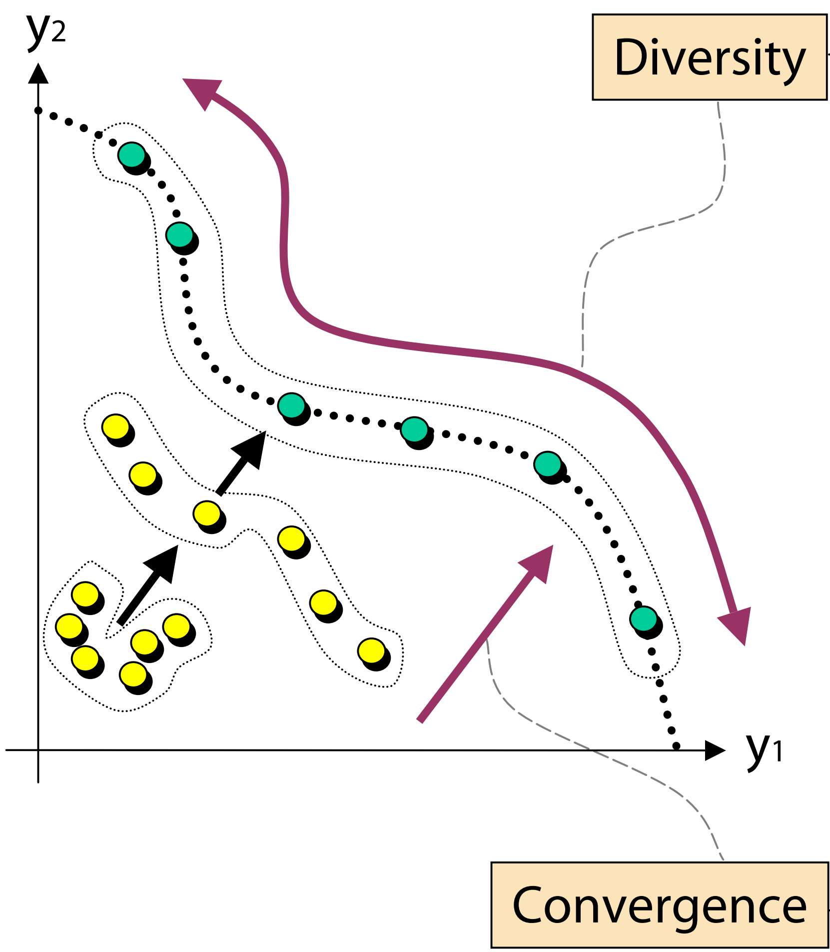
❺ **The Challenge:**

Standard interface for search algorithms

❻ **The End:**

Conclusions and outlook

Issues in EMO



- How to maintain a diverse Pareto set approximation?

  ❷ **density estimation**

- How to prevent nondominated solutions from being lost?

  ❸ **environmental selection**

- How to guide the population towards the Pareto set?

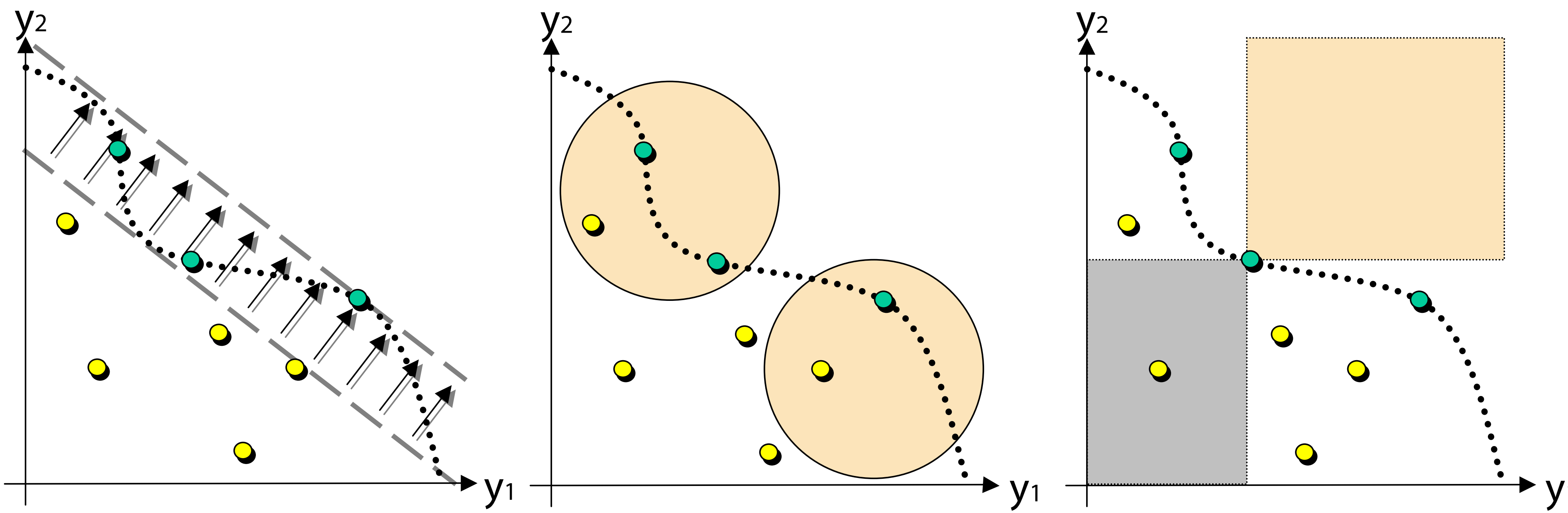
  ❶ **mating selection**

Fitness Assignment Strategies

aggregation-based
weighted sum

criterion-based
VEGA

dominance-based
SPEA2



parameter-oriented
scaling-dependent

set-oriented
scaling-independent

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:

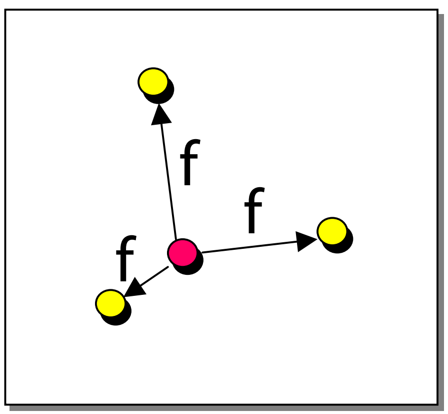
• *MOGA, NPGA*	dominance rank
• *NSGA/NSGA-II*	dominance depth
• *SPEA/SPEA2*	dominance count + rank

Density estimation techniques: *[Silverman: 1986]*

Kernel

MOGA

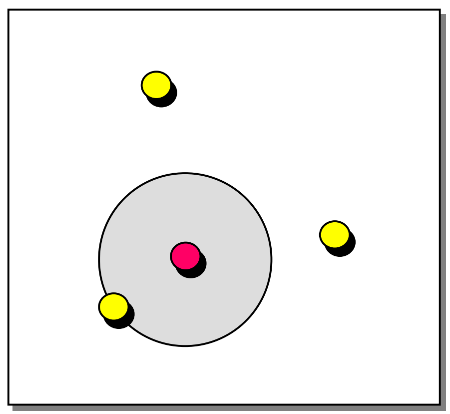
density estimate
=
sum of f values where f is a function of the distance



Nearest neighbor

SPEA2

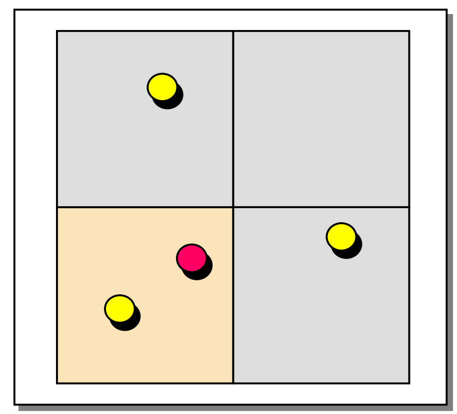
density estimate
=
volume of the sphere defined by the nearest neighbor



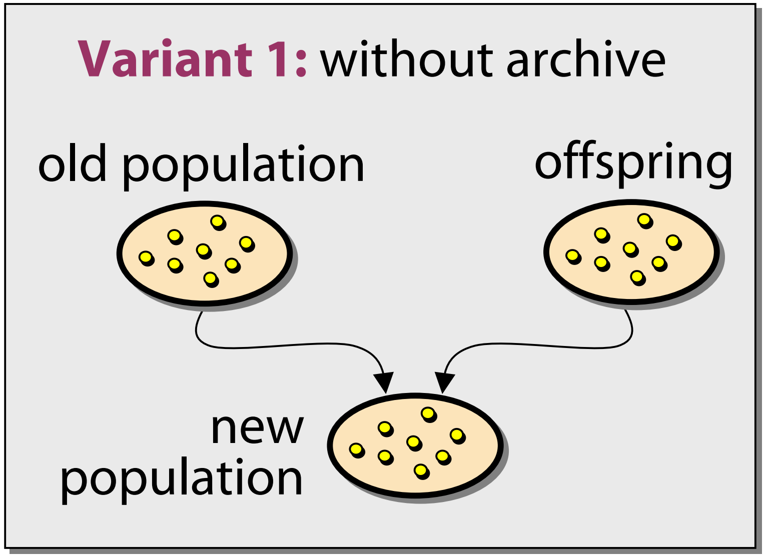
Histogram

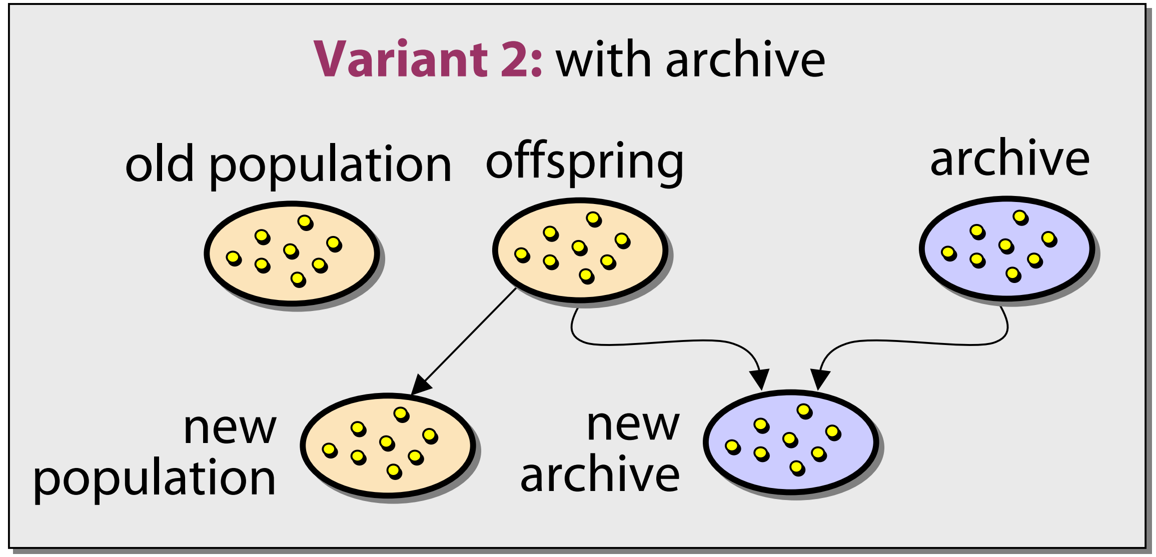
PAES

density estimate
=
number of solutions in the same box



Environmental Selection





Selection criteria:

- **Dominance:** only nondominated solutions are kept
- **Density**: less crowded regions are preferred to crowded regions
- **Time**: old archive members are preferred to new solutions
- **Chance:** each solution has the same probability to enter the archive

Overview

❶ **The Big Picture:**

Optimization and evolutionary computation

❷ **The Construction Kit:**

Design issues and algorithmic concepts

❸ **The Pieces Put Together:**

Example of an algorithm variant

❹ **The Big Question:**

Performance of evolutionary algorithms

❺ **The Challenge:**

Standard interface for search algorithms

❻ **The End:**

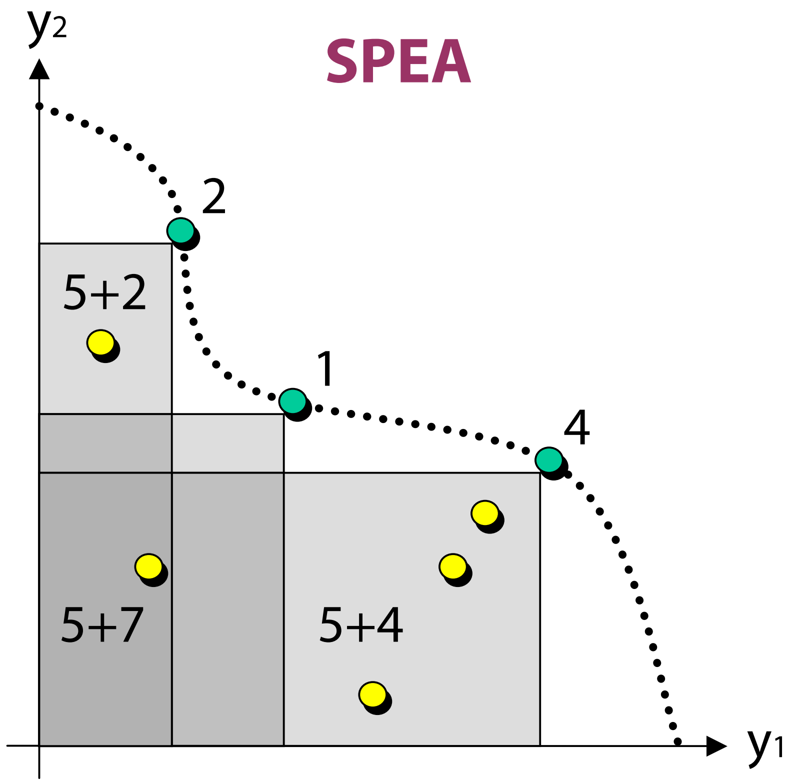
Conclusions and outlook

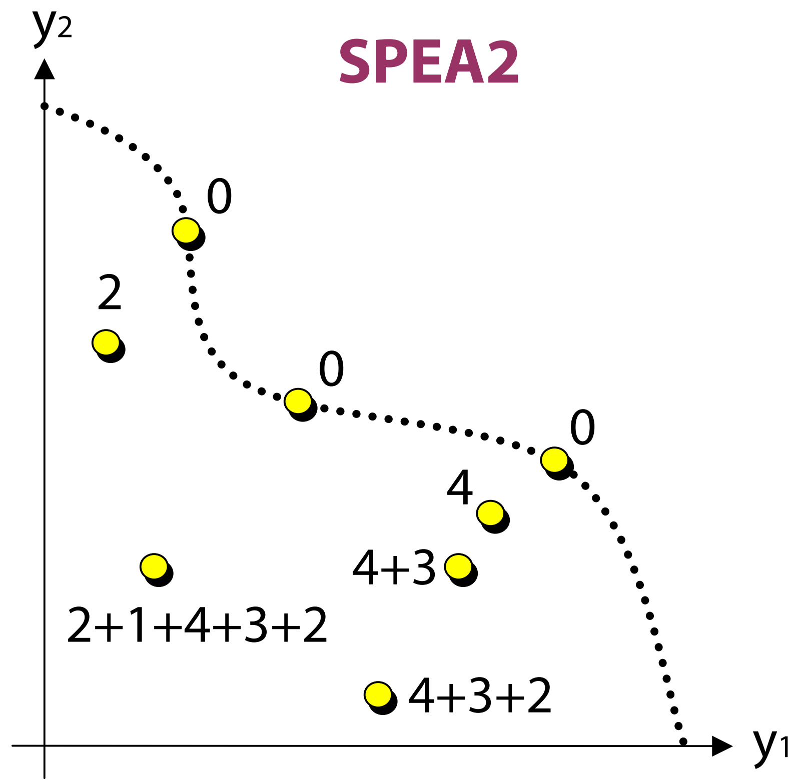
SPEA2 Algorithm

Step 1:	Generate initial population P_0 and empty archive (external set) A_0. Set $t = 0$.
Step 2:	Calculate fitness values of individuals in P_t and A_t.
Step 3:	A_{t+1} = nondominated individuals in P_t and A_t. If size of $A_{t+1} > N$ then reduce A_{t+1}, else if size of $A_{t+1} < N$ then fill A_{t+1} with dominated individuals in P_t and A_t.
Step 4:	If $t > T$ then output the nondominated set of A_{t+1}. Stop.
Step 5:	Fill mating pool by binary tournament selection with replacement on A_{t+1}.
Step 6:	Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Set $t = t + 1$ and go to Step 2.

[Zitzler, Laumanns, Thiele: 2001]

Pareto Fitness Assignment





SPEA

- S (strength) = #dominated solutions
- R (raw fitness) = N + $\sum$ strengths of dominators

SPEA2

- S (strength) = #dominated solutions
- R (raw fitness) = $\sum$ strengths of dominators

Diversity Preservation

Density Estimation

k-th nearest neighbor method:

- Fitness = R + $\underbrace{1 / (2 + D_k)}_{< 1}$
- D_k = distance to the k-th nearest individual
- $k = \sqrt{\text{popsize} + \text{archivesize}}$

Truncation

Incremental approach:

- Remove individual A for which A $<_d$ B for all individuals B
- B $<_d$ A iff:
  - D_k identical for A and B for all k
  - D_k of A greater than D_k of B for a particular k and identical for smaller k

Overview

❶ **The Big Picture:**
Optimization and evolutionary computation

❷ **The Construction Kit:**
Design issues and algorithmic concepts

❸ **The Pieces Put Together:**
Example of an algorithm variant

❹ **The Big Question:**
Performance of evolutionary algorithms

❺ **The Challenge:**
Standard interface for search algorithms

❻ **The End:**
Conclusions and outlook

Performance Assessment: Approaches

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

- Limit behavior

  "Is the Pareto set found, if there are unlimited run-time resources?"

- Run-time analysis

  "How long does it take to generate the Pareto set with high probability?"

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

Limit Behavior

Basic assumptions:

- Every solution can be generated from every other solution by mutation
- The number of iterations t goes to infinity ($t \longrightarrow \infty$)

Studies:

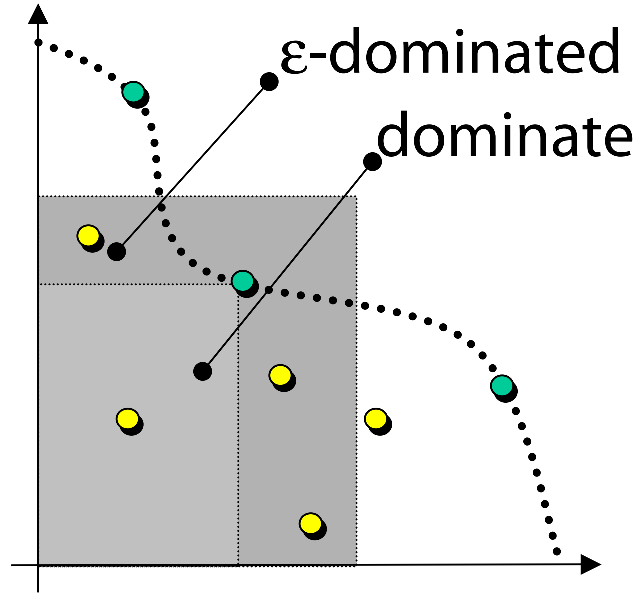
- Convergence: *[Hanne: 1999][Rudolph, Agapie: 2000]*
- Diversity: e.g., *[Knowles, Corne: 2000][Deb et al.: 2001]*
- Convergence + diversity:
  - Unlimited memory resources *[Rudolph and Agapie: 2000]*
  - Limited memory resources *[Laumanns et al.: 2002]*

Epsilon Dominance

Definition 1: ε-Dominance

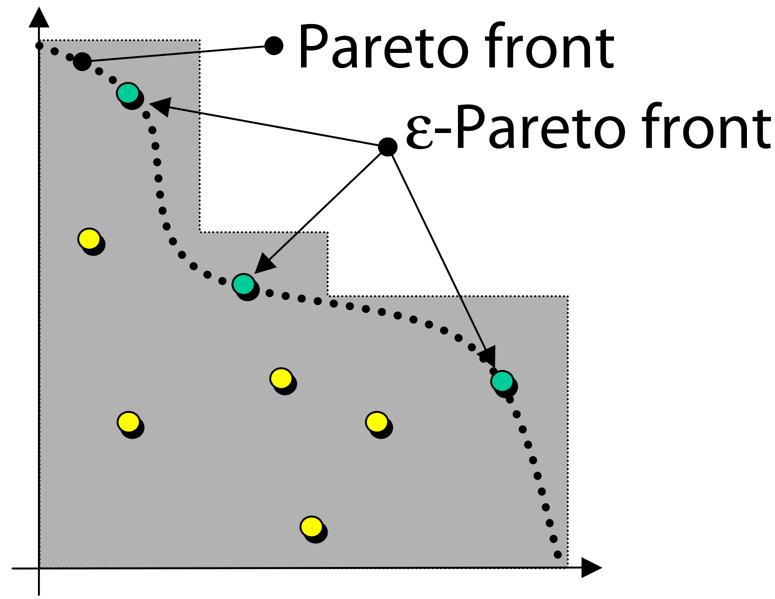
A ε-dominates B iff $\varepsilon \cdot f(A) \geq f(B)$

(known since 1987)



Definition 2: ε-Pareto set

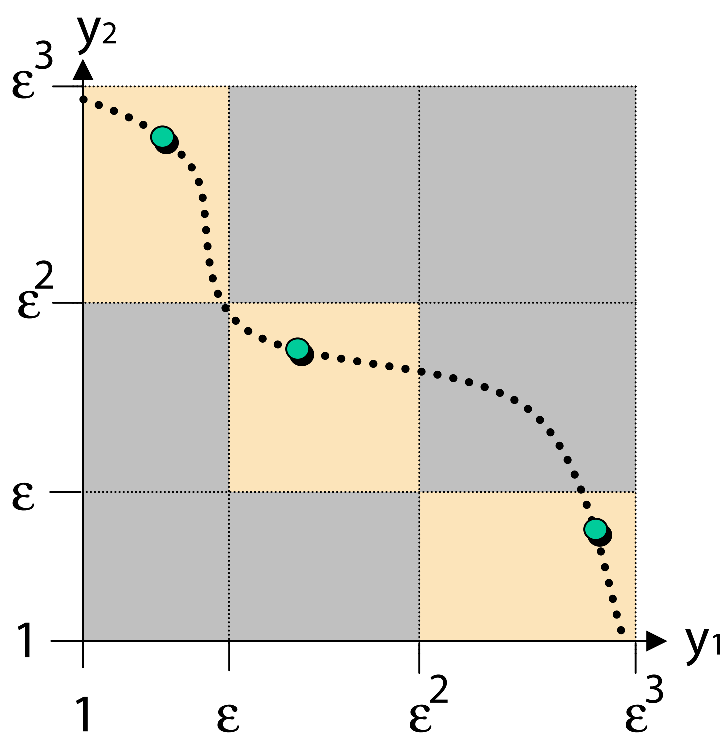
subset of the Pareto set which ε-dominates all Pareto-optimal solutions



Achieving Convergence and Diversity

Goal: Maintain ε-Pareto set

Idea: ε-grid, i.e. maintain a set of nondominated boxes (one solution per box)



Algorithm: (ε-update)

Accept a child if

❶ the corresponding box is not dominated by any box that contains an individual

AND

❷ any other individual in the same box is dominated by the new solution

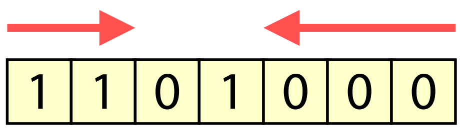
Run-Time Analysis of a Multiobjective EA

Basic question: *[Laumanns et al.: 2002]*

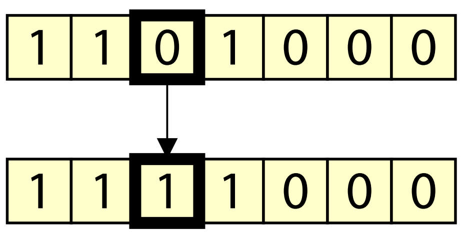
What is the worst case run-time of a multiobjective EA to find the Pareto set with high probability?

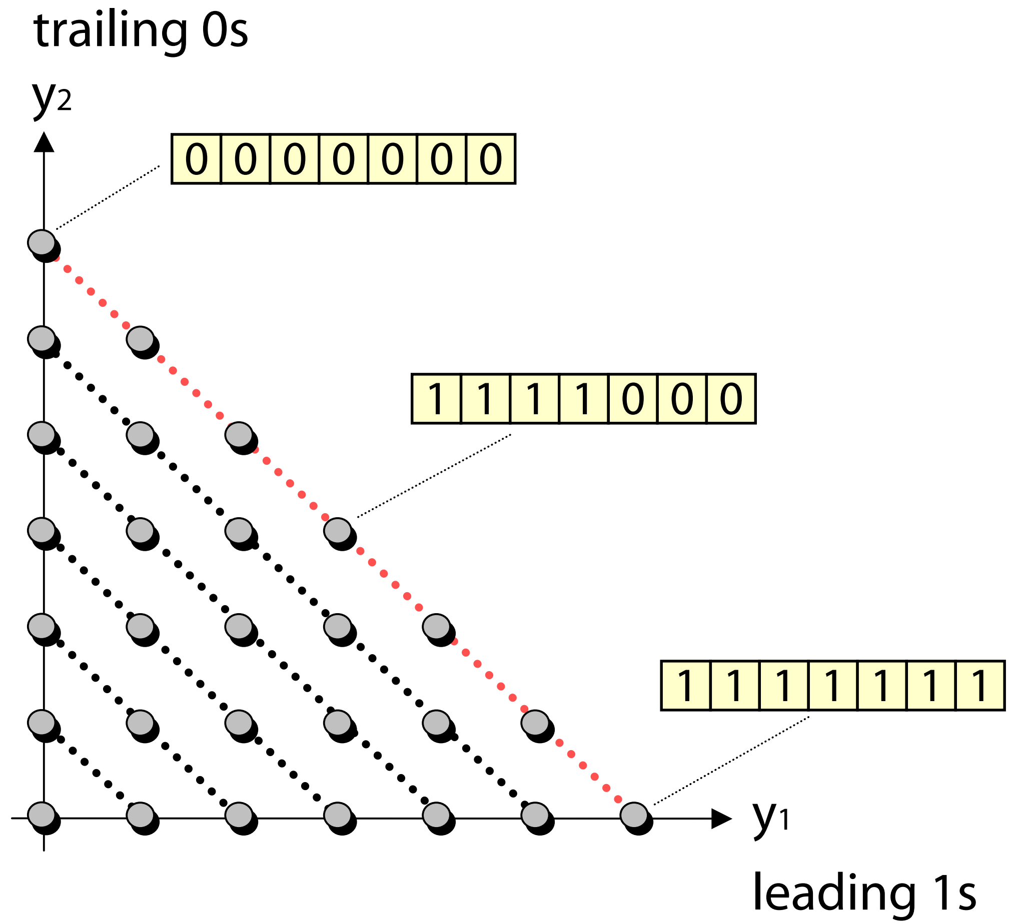
Scenario:

- **Problem:** leading ones, trailing zeros (LOTZ)

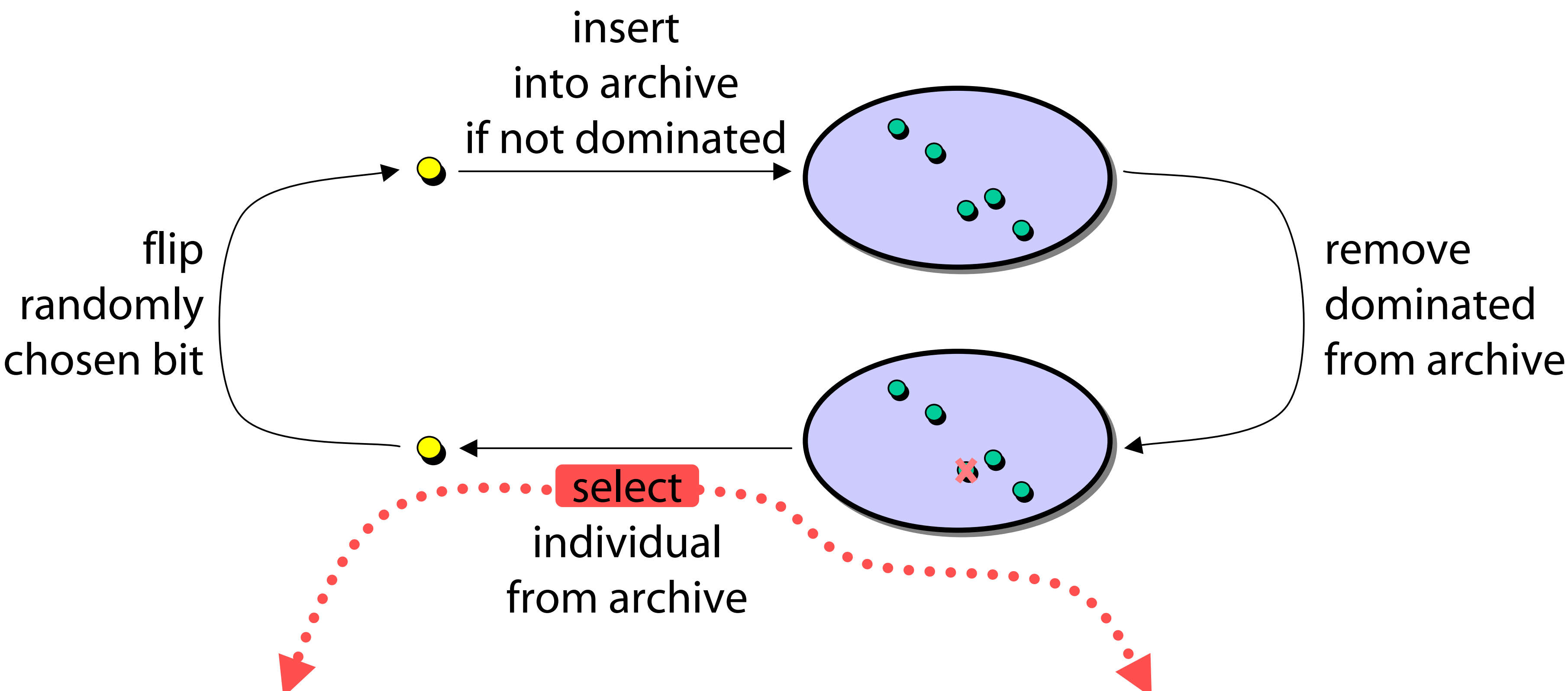


- **Variation:** single point mutation





Two Simple Multiobjective EAs



Variant 1: SEMO

Each individual in the archive is selected with the same probability (uniform selection)

Variant 2: FEMO

Select individual with minimum number of children (fair selection)

Results of Analysis

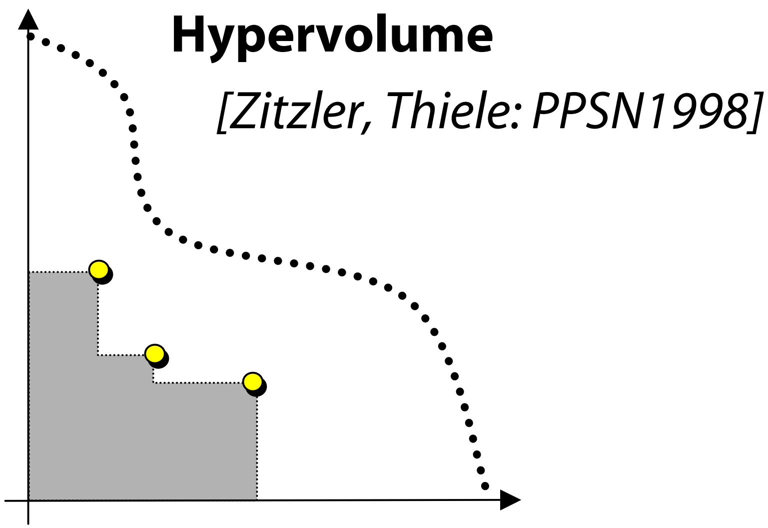
- Simple multiobjective EA with uniform selection (SEMO): $\Theta(n^3)$
  - To get to the Pareto front requires n^2 steps
  - To cover the entire front needs n^3 steps
- Simple multiobjective EA with fair selection (FEMO): $\Theta(n^2 \log n)$
  - Fair selection helps to spread over the Pareto front
- Multistart single-objective optimizer: $\Omega(n^3)$
  - In average, one out of *n* mutations successful
  - To get to the Pareto front, *n* successful mutations needed
  - Overall *n* Pareto-optimal solutions have to be found

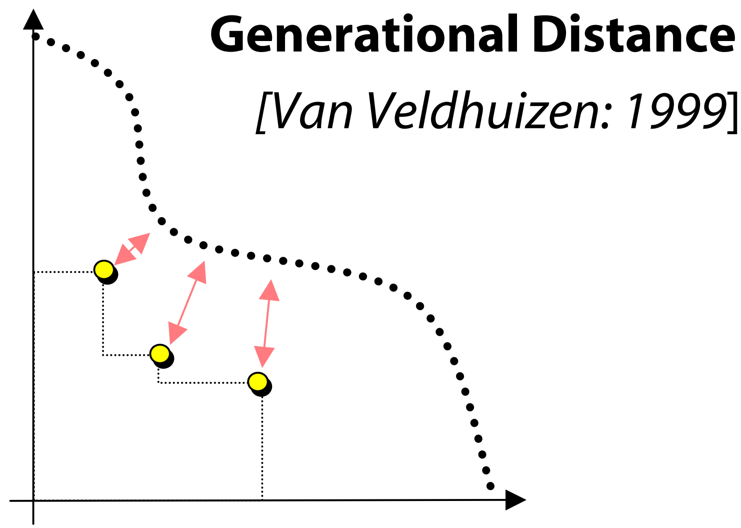
multiobjective EA faster than multistart strategy

Empirical Performance Assessment

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

Popular approach: unary quality measures

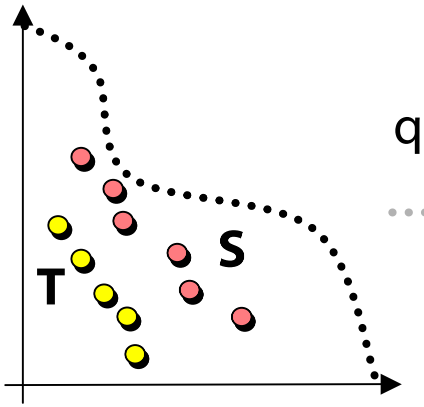




- Assign each outcome a *real number*
- Outcomes are compared by comparing the corresponding *values*

Quality Measures: Theoretical Results I

Basic question: Can we say on the basis of the quality measures *whether* or *that* an algorithm outperforms another?



application of quality measures

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

There is no combination of unary quality measures such that **S** is better than **T** in all criteria is equivalent to **S** dominates **T**

Unary quality measures usually do not tell that **S** dominates **T**; at maximum that **S** does not dominate **T**

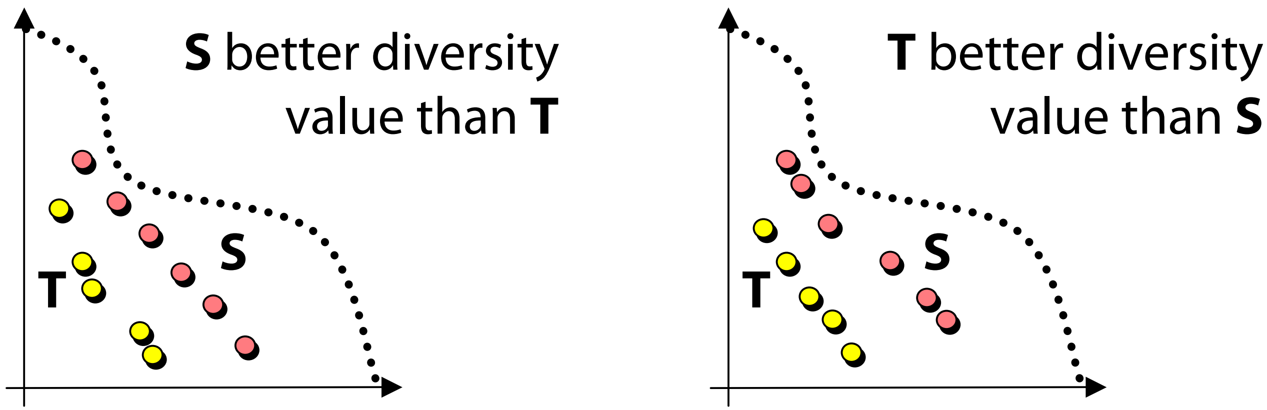
[Zitzler et al.: 2002]

Quality Measures: Theoretical Results II

Many popular quality measures are not compliant with the dominance relation

[Hansen, Jaszkiewicz: 1998][Knowles, Corne: 2002][Zitzler et al.: 2002]

Example: diversity measures

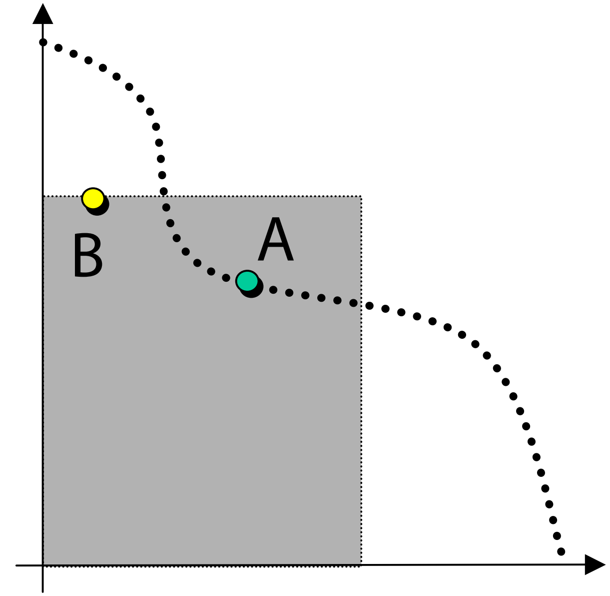


Needed: appropriate binary quality measures that indicate *whether* an outcome dominates another, e.g., ε-measure

Epsilon Quality Measure

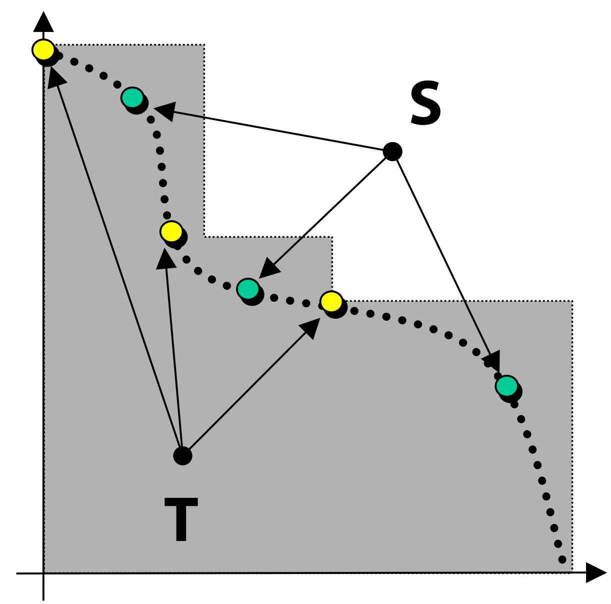
Definition 3: single solutions

$I_\varepsilon(A,B)$ = minimum ε such that A ε-dominates B



Definition 4: sets of solutions

$I_\varepsilon(\mathbf{S},\mathbf{T})$ = minimum ε such that each solution in **T** is ε-dominated by at least one solution in **S**



[Zitzler et al.: 2002]

Overview

❶ **The Big Picture:**
Optimization and evolutionary computation

❷ **The Construction Kit:**
Design issues and algorithmic concepts

❸ **The Pieces Put Together:**
Example of an algorithm variant

❹ **The Big Question:**
Performance of evolutionary algorithms

❺ **The Challenge:**
Standard interface for search algorithms

❻ **The End:**
Conclusions and outlook

Implementation: Current Situation

Application engineer

- knowledge in the algorithm domain necessary
- state-of-the-art algorithms get more and more complex
- many algorithms

Algorithm designer

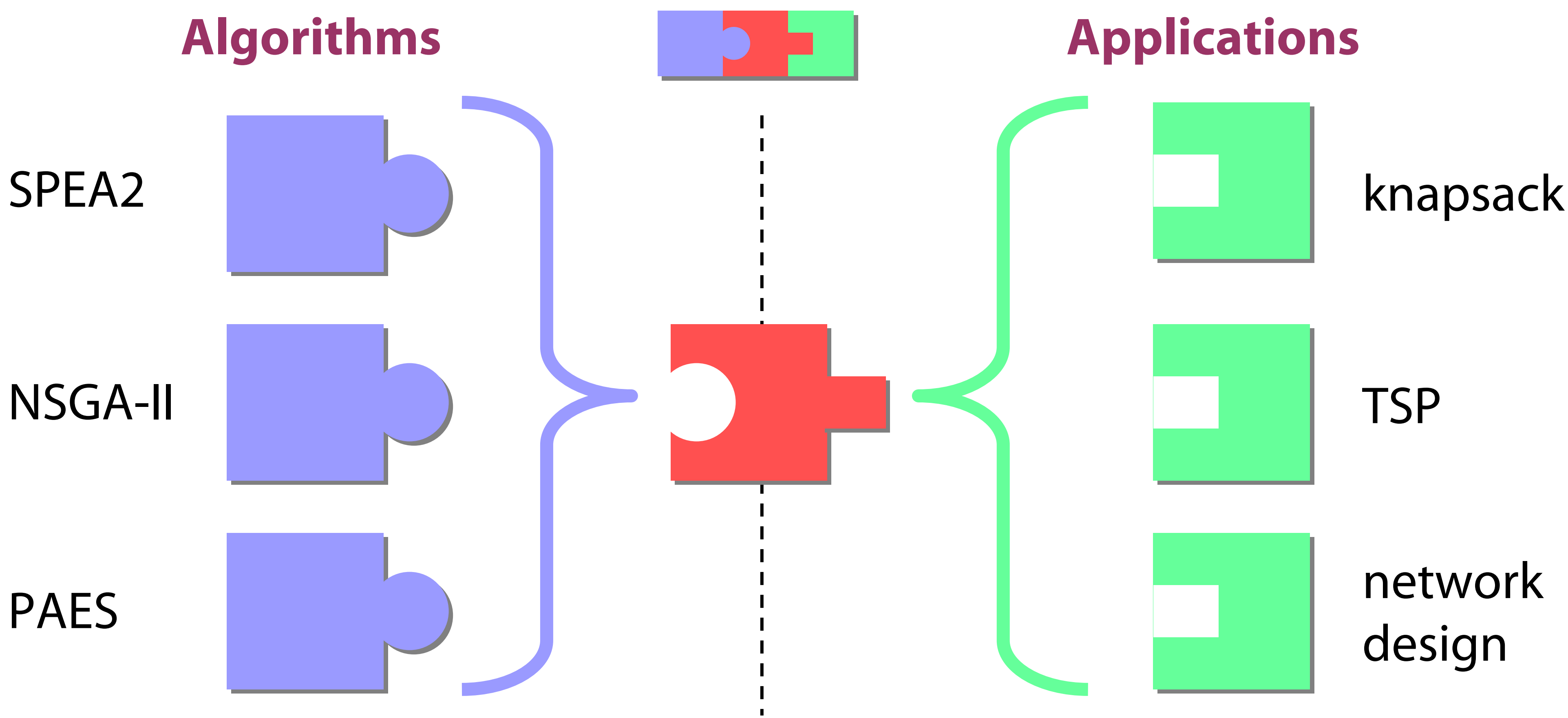
- comparison to competing algorithms mandatory
- tests on various benchmark problems necessary
- algorithms and applications become increasingly complex

high implementation effort / risk of implementation errors

Programming libraries:

+ valuable tools to tailor a particular technique to a specific application
− exchange of optimization algorithm or application still difficult

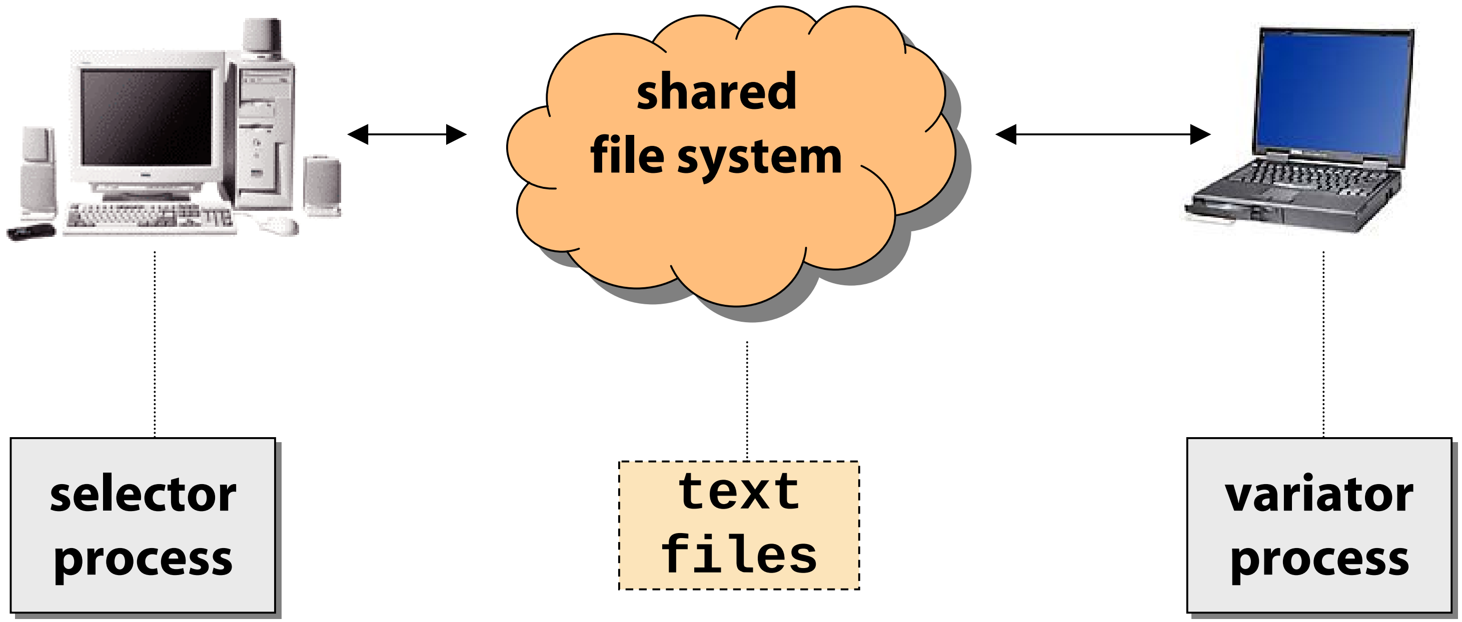
The Concept of PISA



text-based

Platform and programming language independent **I**nterface for **S**earch **A**lgorithms *[Bleuler et al.: 2002]*

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

Conclusions

Why using an evolutionary algorithm?

- **Flexibility:** problem formulation can be easily modified / extended (minimum requirements)
- **Multiple objectives:** the solution space can be explored in a single optimization run
- **Feasibility:** EAs are applicable to complex and huge search spaces

Why multiobjective optimization?

- **Robustness:** aggregation of several objectives into a single one requires setting of parameters
- **Confidence:** it is easier to select a solution if alternatives are known

Main application of EMO: design space exploration

More on EMO

Links:

- EMO mailing list:
  http://w3.ualg.pt/lists/emo-list/
- EMO bibliography:
  http://www.lania.mx/~ccoello/EMOO/
- PISA website:
  http://www.tik.ee.ethz.ch/pisa/

Events:

- Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), April 8-11, 2003, Algarve, Portugal:
  http://conferences.ptrede.com/emo03/

Acknowledgments:

- Stefan Bleuler, Marco Laumanns, Lothar Thiele