

# Issues in Design Space Exploration Using Randomized Search Algorithms

**Eckart Zitzler**

Computer Engineering (TIK), ETH Zurich, Switzerland  
Workshop on Distributed Embedded Systems, 11/22/2005



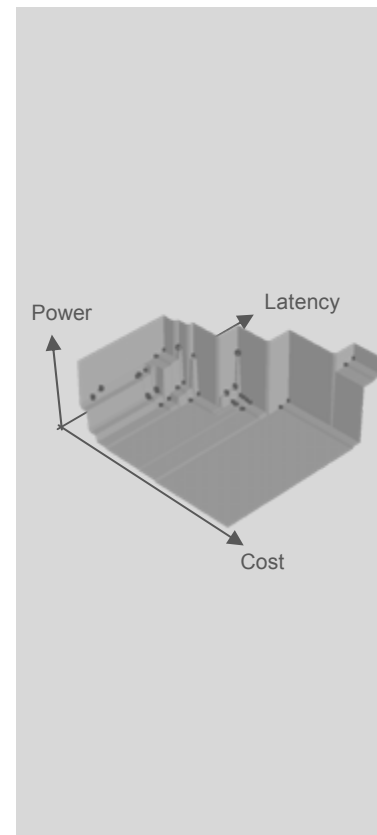
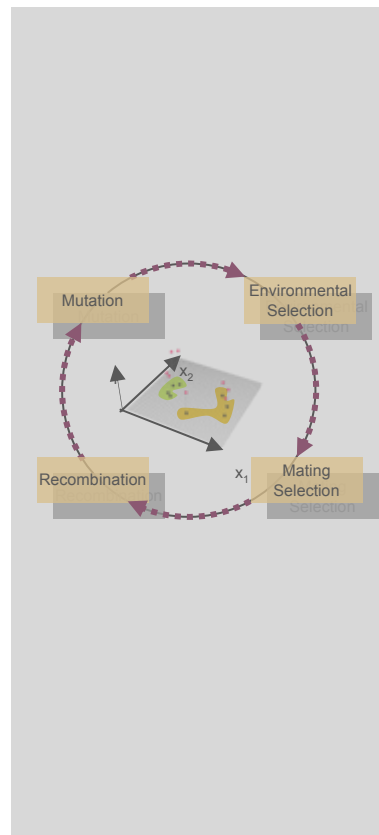
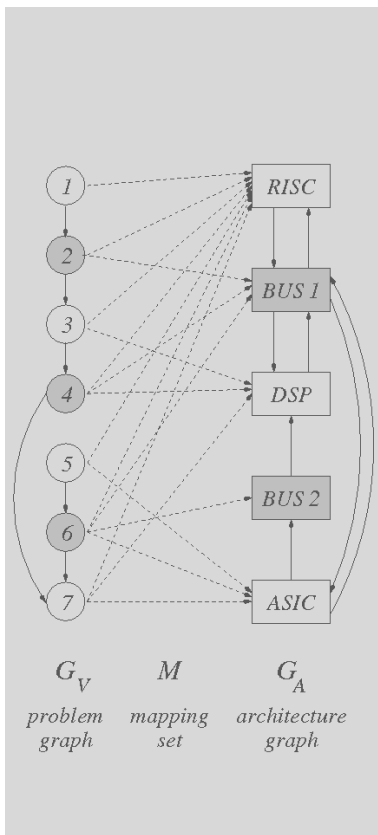
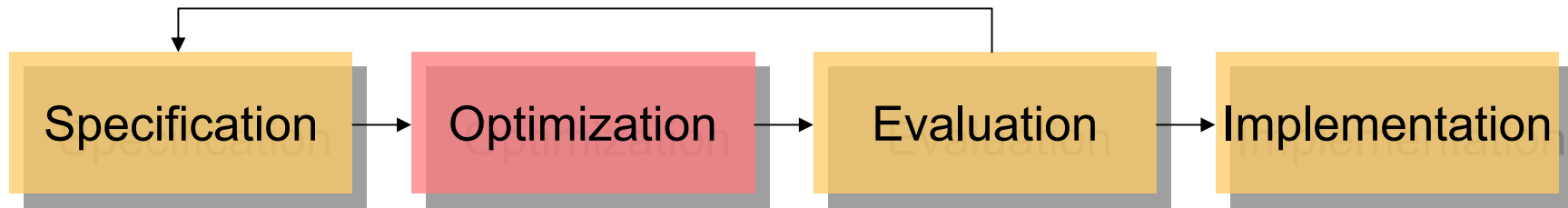
**ETH**

Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

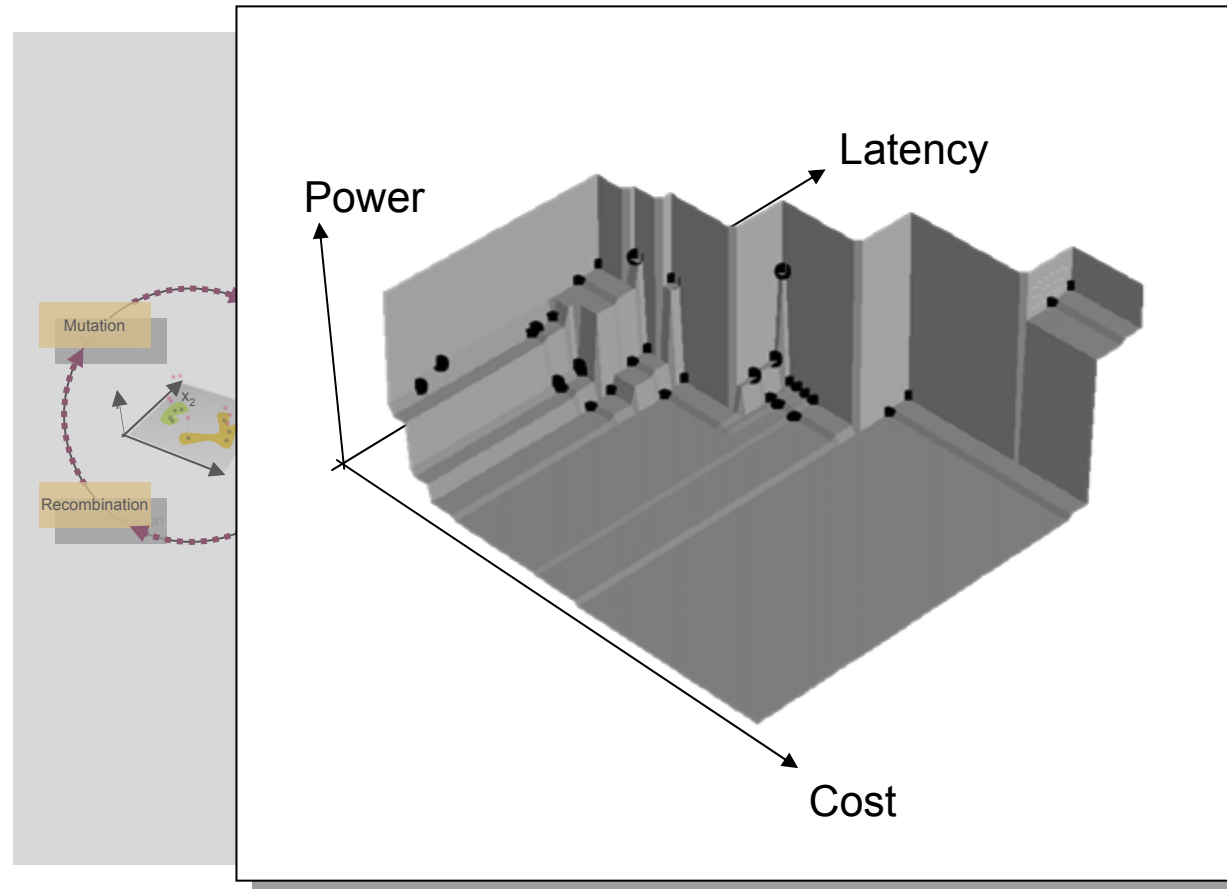
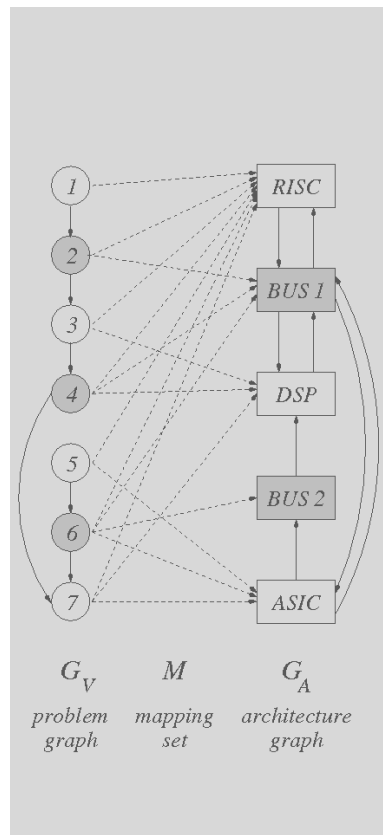
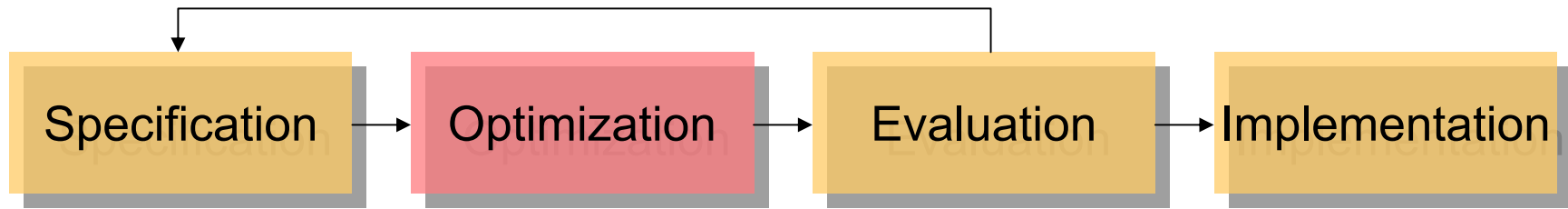
**TIK**

Computer Engineering  
and Networks Laboratory

# Design Space Exploration: Setting

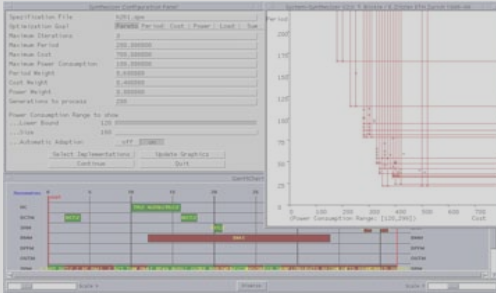


# Design Space Exploration: Setting



# Example Applications

## Architecture Synthesis for Embedded Systems



## Network Processor Design

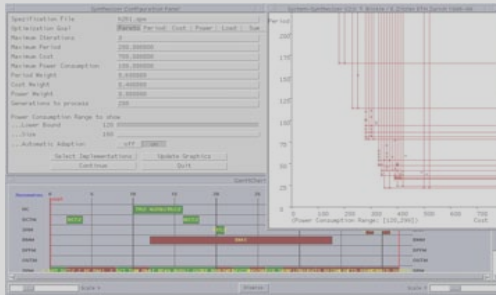


## Software Synthesis for DSPs



# Example Applications

## Architecture Synthesis for Embedded Systems



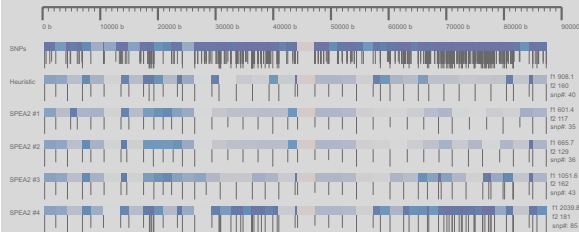
## Network Processor Design



## Software Synthesis for DSPs

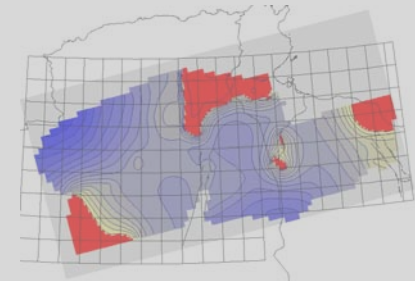


## Experiment Design for Genotyping



## Optimal Usage of Non-renewable Groundwater Resources in the Sahara

Tobias Siegfried  
Prof. Dr. Wolfgang Kinzelbach  
Institut für Hydromechanik und Wasserwirtschaft (IHW)

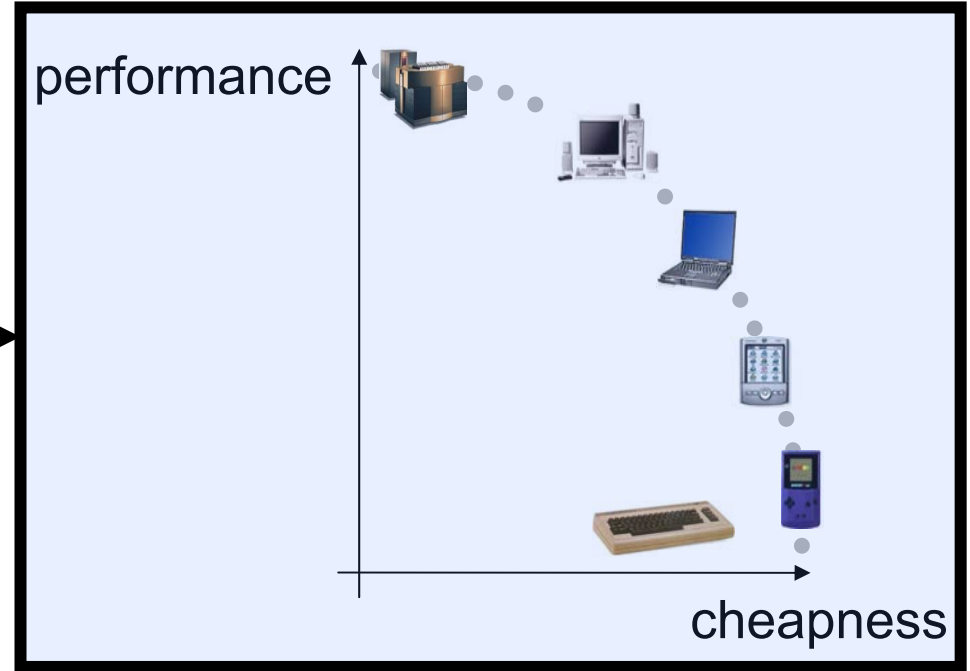
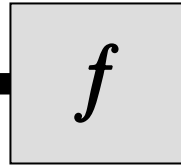


# Global Optimization

design space

objectives

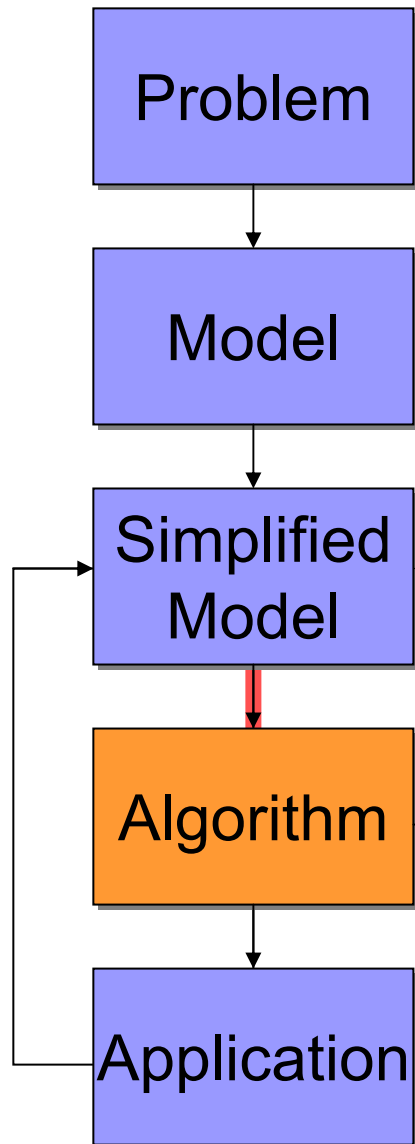
objective space



## Challenges:

- complexity of the solution space
- conflicting optimization criteria
- uncertain information
- computationally expensive objective functions

# Bottom-Up Approach



**Focus:** a good algorithm...

❶ Design of a problem-specific algorithm using known concepts:

- Dynamic programming
- Branch and bound
- Divide and conquer

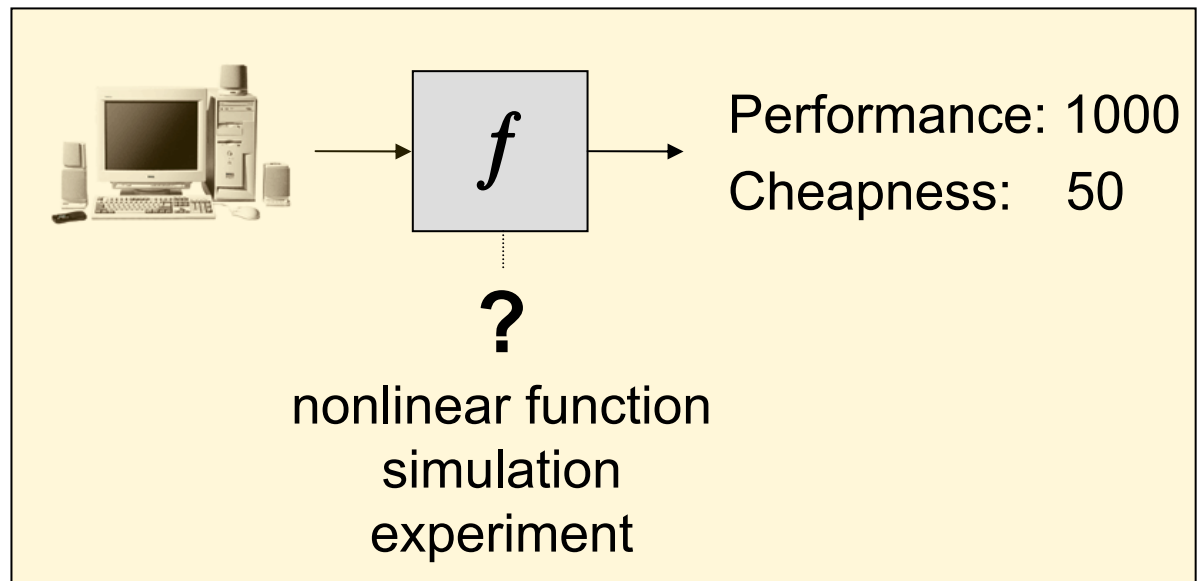
❷ Usage of a general search algorithm with model constraints:

- Integer linear programming
- Gradient methods

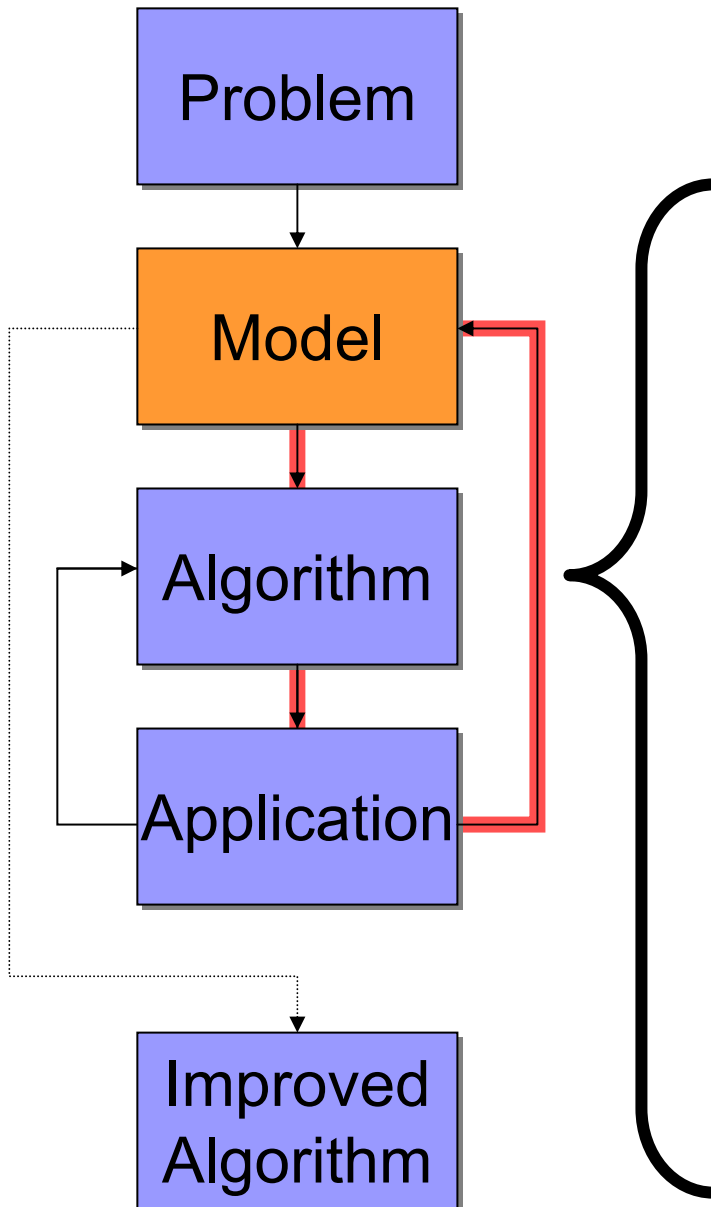
# Top-Down Approach

**Focus:** a good model...

Usage of black-box optimization methods to approximate the optima:



- Evolutionary algorithms
- Simulated annealing

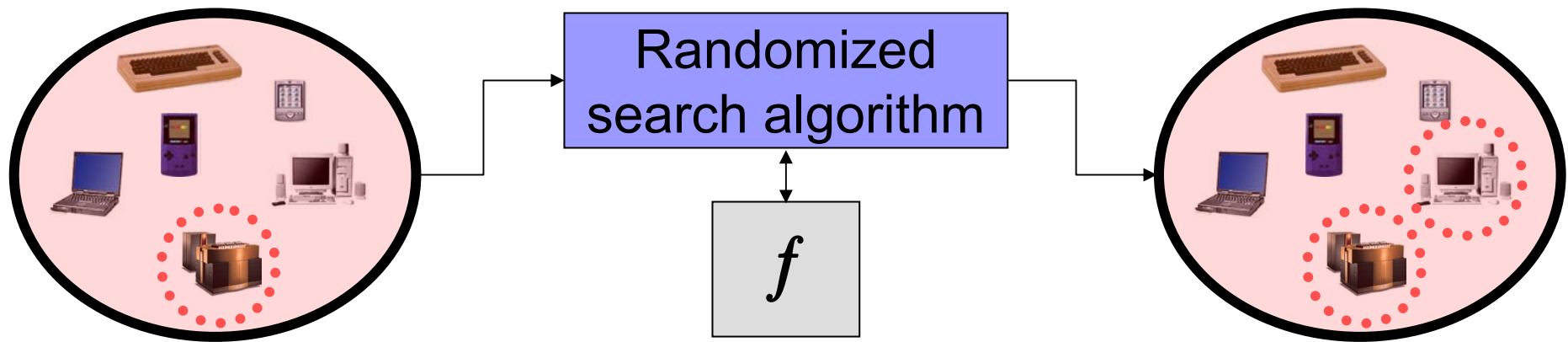




# Randomized Search Algorithms (RSAs)

**Idea:** find good solutions without investigating all solutions

**Assumption:** better solutions can be found in the neighborhood of good solutions



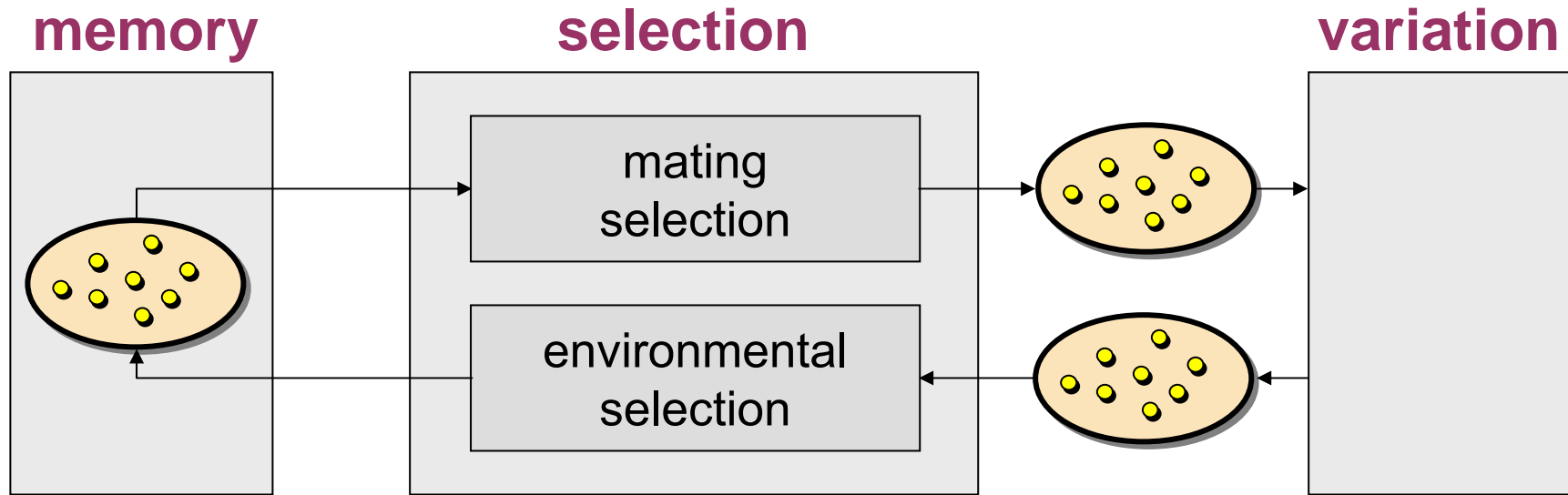
**$t = 1$ :**

(randomly) choose a solution  $x_1$  to start with

**$t \rightarrow t+1$ :**

(randomly) choose a solution  $x_{t+1}$  using solutions  $x_1, \dots, x_t$

# Types of Randomized Search Algorithms



	memory	selection	variation	
<b>EA</b> evolutionary algorithm	$\geq 1$	both	$\geq 1$ $\geq 1$	$N : M$ randomized
<b>TS</b> tabu search	1	no mating selection	1 $\geq 1$	1 : M deterministic
<b>SA</b> simulated annealing	1	no mating selection	1 $\geq 1$	1 : M randomized
<b>ACO</b> ant colony optimization	1	neither	1 1	1 : 1 randomized

# Key Issues in Design Space Exploration Using RSA

## How to...

- handle multiple objectives
- account for uncertainty
- optimize for robustness
- incorporate user preferences
- achieve diversity in the parameter space
- reduce the design space complexity
- statistically assess the quality of the outcome
- simplify implementation and maximize reusability
- deal with limited memory resources
- postprocessing of tradeoff surface

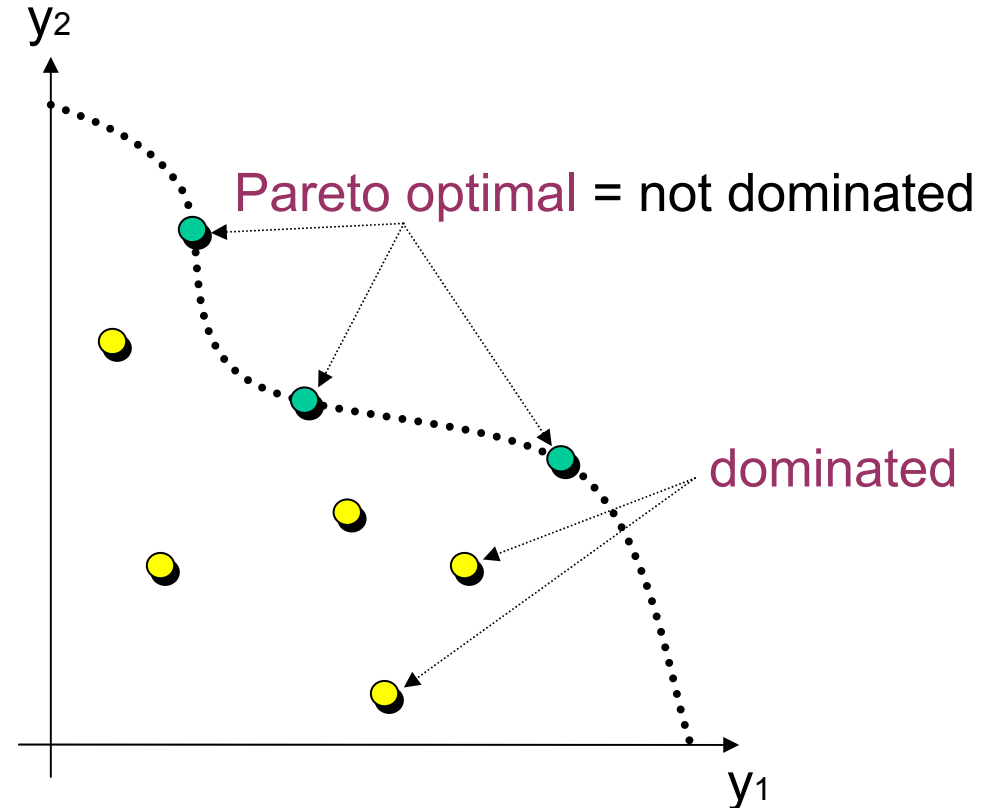
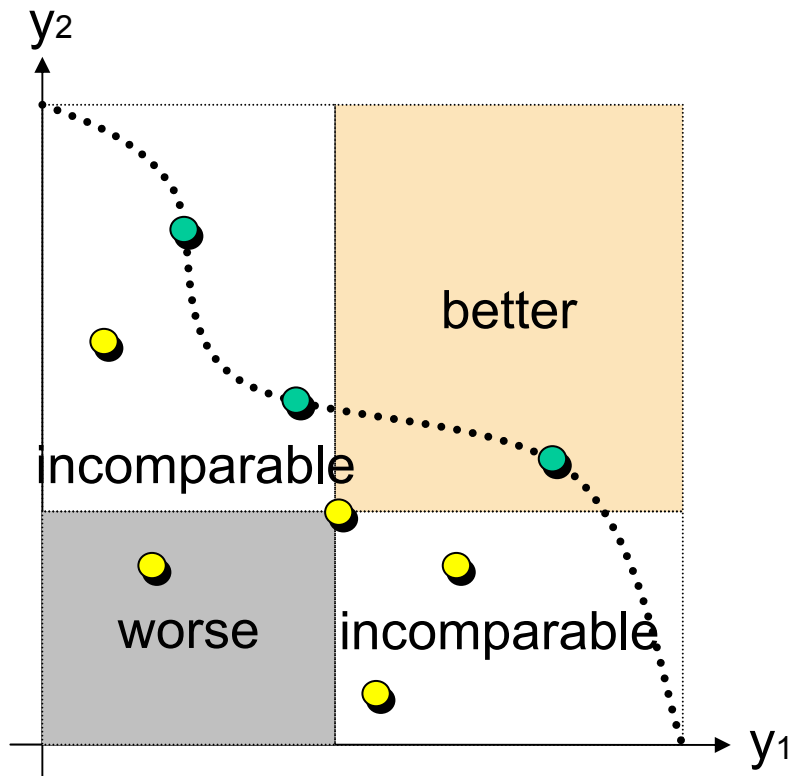
# Key Issues in Design Space Exploration Using RSA

## How to...

- handle multiple objectives
- account for uncertainty
- optimize for robustness
- incorporate user preferences
- achieve diversity in the parameter space
- reduce the design space complexity
- statistically assess the quality of the outcome
- simplify implementation and maximize reusability
- deal with limited memory resources
- postprocessing of tradeoff surface

# The Multiobjective Scenario

Maximize  $(y_1, y_2, \dots, y_n) = (f_1(x_1, x_2, \dots, x_k), \dots, f_n(x_1, x_2, \dots, x_k))$

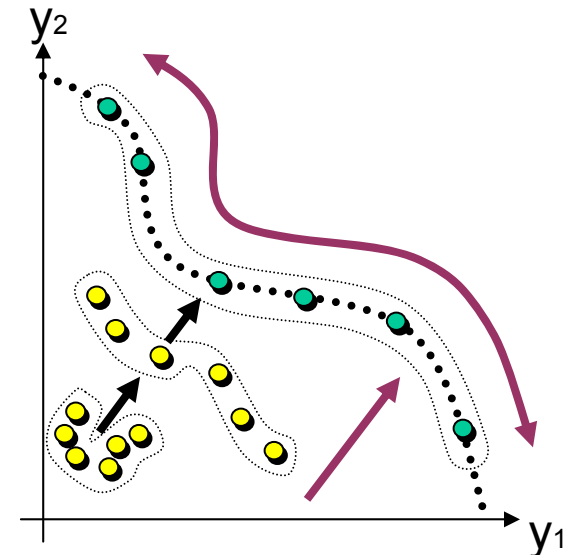


Problem is underdetermined...

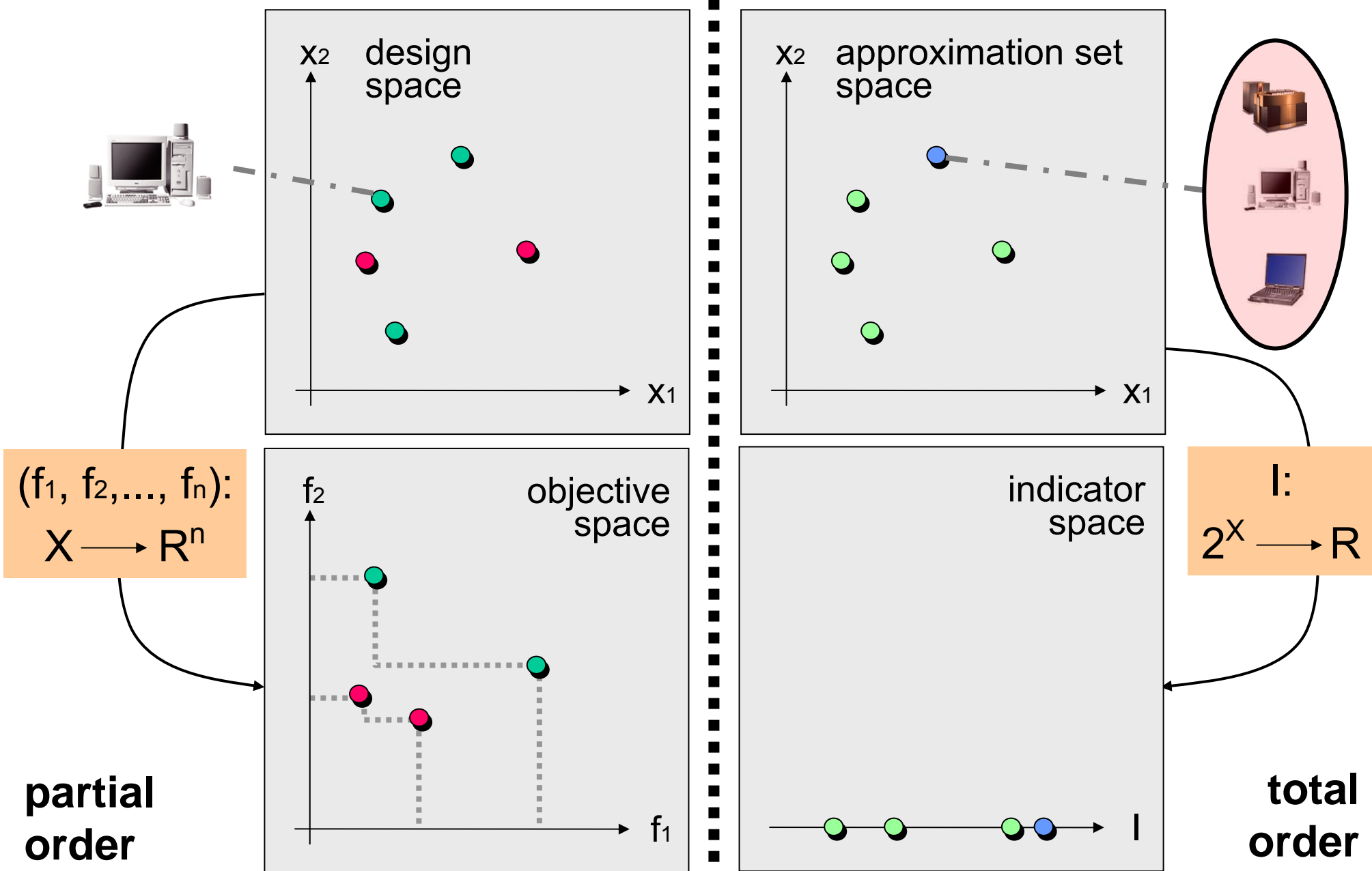
# What Is the Optimization Goal?

- 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:
    - ① close to the Pareto front
    - ② well distributed

*[Deb:01]*



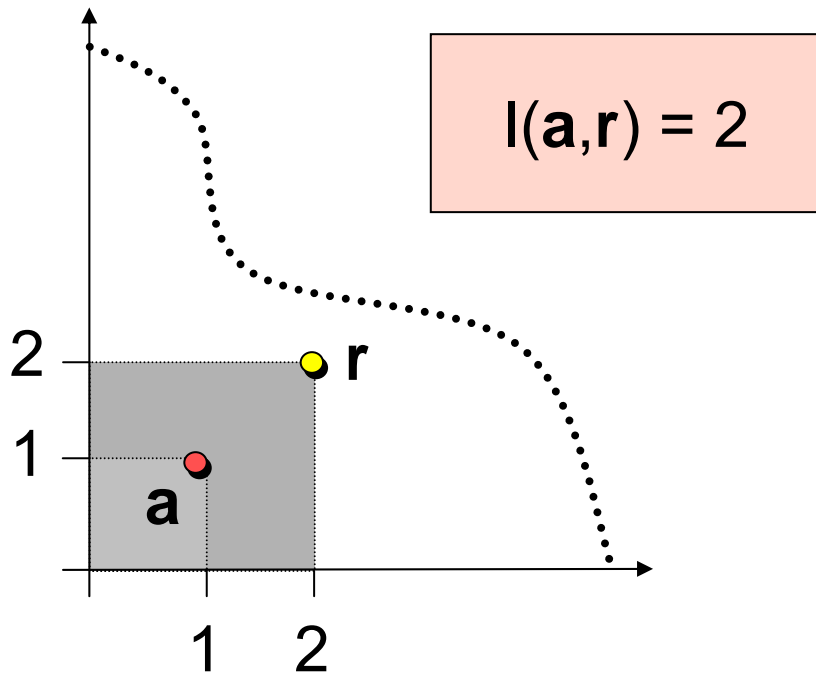
# The Actual Problem



# The $\varepsilon$ -Quality Indicator

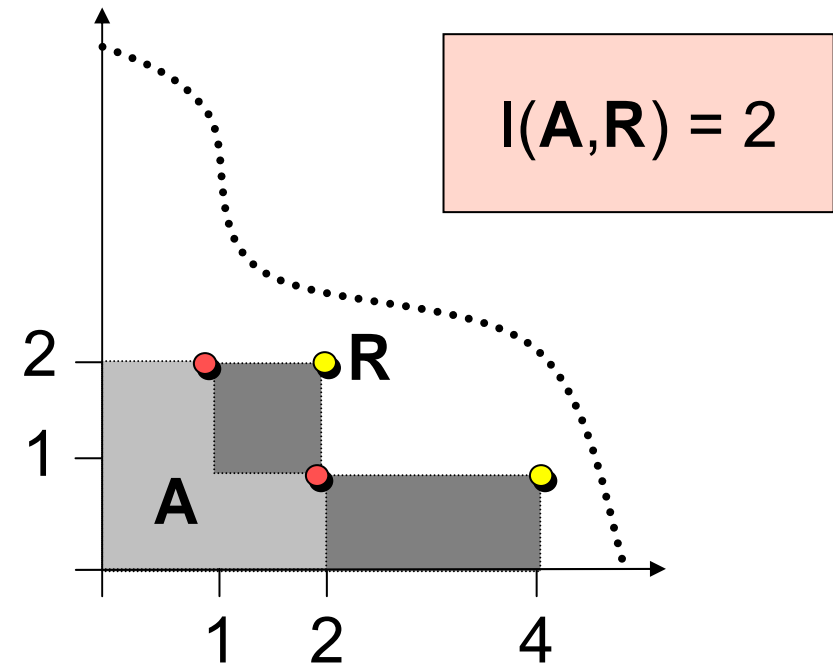
Two solutions:

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



Two approximations:

$$I(\mathbf{A}, \mathbf{R}) = \max_{\mathbf{r} \in \mathbf{R}} \min_{\mathbf{a} \in \mathbf{A}} I(\mathbf{a}, \mathbf{r})$$

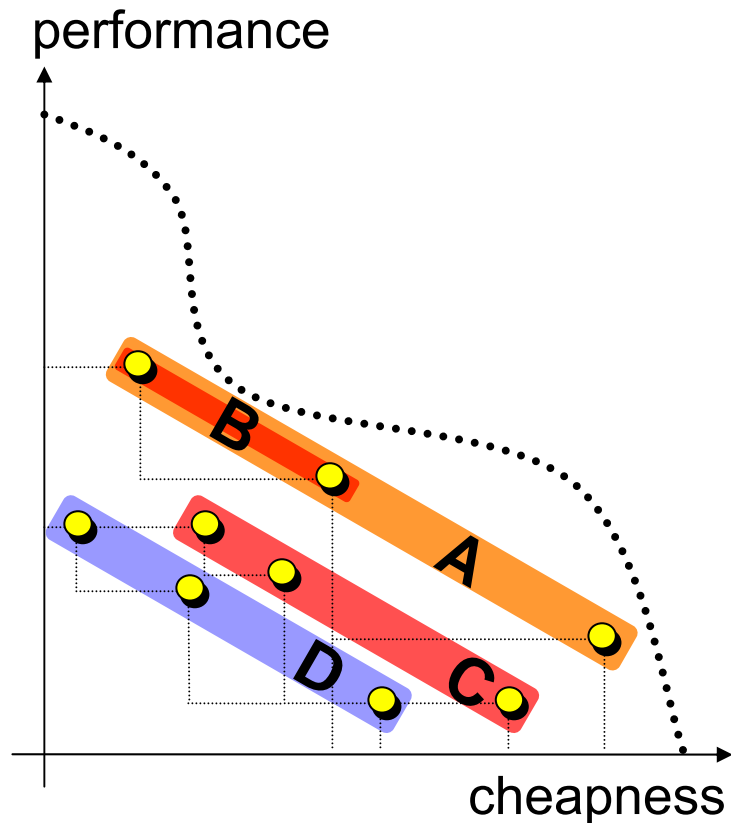


$I(\mathbf{A}, \mathbf{R}) =$  minimum factor by which  $\mathbf{A}$  needs to be “improved” such the (fixed) reference set  $\mathbf{R}$  is entirely covered



# Pareto Compliance

Indicators should be **Pareto compliant** =  
the order induced by I should be an extension of the  
order induced by  $f_1, \dots, f_n$



**Pareto dominance:**

A better than B

B better than C

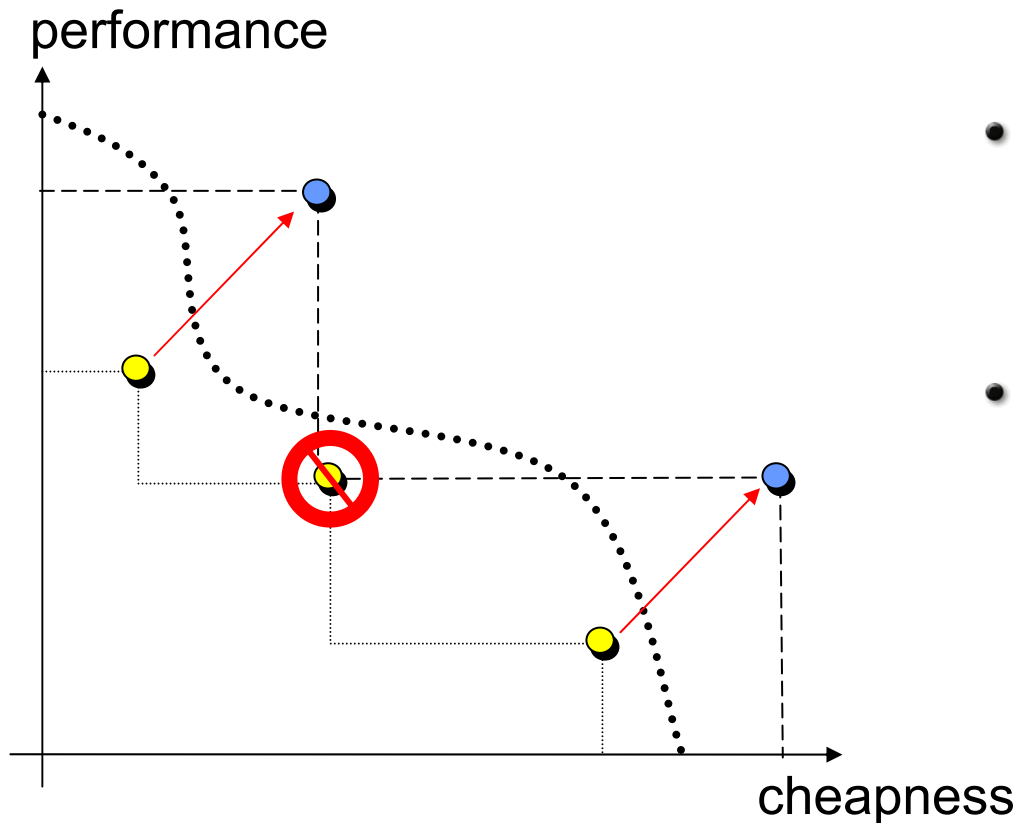
C better than D

**Indicator ranking:**

$I(A) \geq I(B) \geq I(C) \geq I(D)$

# Indicator-Based Selection: Main Idea

**Fitness  $x$**  = loss in indicator value if  $x$  is removed  
=  $I(\mathbf{A}-\{x\}, \mathbf{A})$   
=  $I(\mathbf{A}-\{x\}, \{x\})$



- “optimal” in steady state (one solution per iteration)
- needs to be extended to break ties

# Implementation for the $\varepsilon$ -Quality Indicator

**Fitness  $\mathbf{x}$**  = loss in indicator value if  $\mathbf{x}$  is removed  
=  $I(\mathbf{A}-\{\mathbf{x}\}, \{\mathbf{x}\})$   
=  $\min_{\mathbf{a} \in \mathbf{A}-\{\mathbf{x}\}} \{ I(\{\mathbf{a}\}, \{\mathbf{x}\}) \}$

**More precisely:**

Fitness vector = sorted pairwise indicator values

**Fast approximation:**

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

# Empirical Validation

		SPEA 2		NSGA-II		SPEA2 <sub>adap</sub>		IBEA <sub>ε, adap</sub>	
		P value	T	P value	T	P value	T	P value	T
ZDT6	NSGA-II	$5.6073 \cdot 10^{-4}$	↑						
	SPEA2 <sub>adap</sub>	> 5%	≡	$8.1975 \cdot 10^{-6}$	↓				
	IBEA <sub>ε, adap</sub>	$8.1014 \cdot 10^{-9}$	↑	$2.0023 \cdot 10^{-9}$	↑	$1.9568 \cdot 10^{-9}$	↑		
	IBEA <sub>HD, adap</sub>	0.0095	↑	> 5%	≡	$5.4620 \cdot 10^{-9}$	↑	$1.3853 \cdot 10^{-9}$	↓
DTLZ2	NSGA-II	$3.0199 \cdot 10^{-10}$	↓						
	SPEA2 <sub>adap</sub>	> 5%	≡	$3.0199 \cdot 10^{-10}$	↑				
	IBEA <sub>ε, adap</sub>	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑		
	IBEA <sub>HD, adap</sub>	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$5.5329 \cdot 10^{-7}$	↓
DTLZ6	NSGA-II	$8.1014 \cdot 10^{-9}$	↓						
	SPEA2 <sub>adap</sub>	> 5%	≡	$6.1210 \cdot 10^{-9}$	↑				
	IBEA <sub>ε, adap</sub>	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑		
	IBEA <sub>HD, adap</sub>	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.0199 \cdot 10^{-10}$	↑	$3.5923 \cdot 10^{-4}$	↓
KUR	NSGA-II	> 5%	≡						
	SPEA2 <sub>adap</sub>	> 5%	≡	> 5%	≡				
	IBEA <sub>ε, adap</sub>	$3.0199 \cdot 10^{-10}$	↓	$3.0199 \cdot 10^{-10}$	↓	$6.6955 \cdot 10^{-10}$	↓		
	IBEA <sub>HD, adap</sub>	$3.0199 \cdot 10^{-10}$	↓	$3.0199 \cdot 10^{-10}$	↓	$4.9752 \cdot 10^{-10}$	↓	> 5%	≡
Knap.	NSGA-II	> 5%	≡						
	SPEA2 <sub>adap</sub>	> 5%	≡	> 5%	≡				
	IBEA <sub>ε, adap</sub>	> 5%	≡	> 5%	≡	> 5%	≡		
	IBEA <sub>HD, adap</sub>	> 5%	≡	> 5%	≡	> 5%	≡	> 5%	≡
EXPO2	NSGA-II	> 5%	≡						
	SPEA2 <sub>adap</sub>	> 5%	≡	0.0189	↑				
	IBEA <sub>ε, adap</sub>	$1.0837 \cdot 10^{-8}$	↑	$2.6753 \cdot 10^{-9}$	↑	$6.4048 \cdot 10^{-8}$	↑		
	IBEA <sub>HD, adap</sub>	$1.9638 \cdot 10^{-7}$	↑	$1.2260 \cdot 10^{-8}$	↑	$6.6261 \cdot 10^{-7}$	↑	> 5%	≡
EXPO3	NSGA-II	> 5%	≡						
	SPEA2 <sub>adap</sub>	> 5%	≡	> 5%	≡				
	IBEA <sub>ε, adap</sub>	$4.3165 \cdot 10^{-8}$	↑	$5.0801 \cdot 10^{-8}$	↑	$3.1159 \cdot 10^{-7}$	↑		
	IBEA <sub>HD, adap</sub>	$2.4189 \cdot 10^{-7}$	↑	$1.5732 \cdot 10^{-7}$	↑	$1.1653 \cdot 10^{-8}$	↑	> 5%	≡
EXPO4	NSGA-II	> 5%	≡	-					
	SPEA2 <sub>adap</sub>	> 5%	≡	$9.4209 \cdot 10^{-4}$	↓				
	IBEA <sub>ε, adap</sub>	$1.8546 \cdot 10^{-10}$	↑	$6.9754 \cdot 10^{-10}$	↑	$1.8390 \cdot 10^{-10}$	↑		
	IBEA <sub>HD, adap</sub>	$1.9883 \cdot 10^{-10}$	↑	$1.0221 \cdot 10^{-9}$	↑	$1.9716 \cdot 10^{-10}$	↑	> 5%	≡

significantly better  
than all other  
algorithms

# Key Issues in Design Space Exploration Using RSA

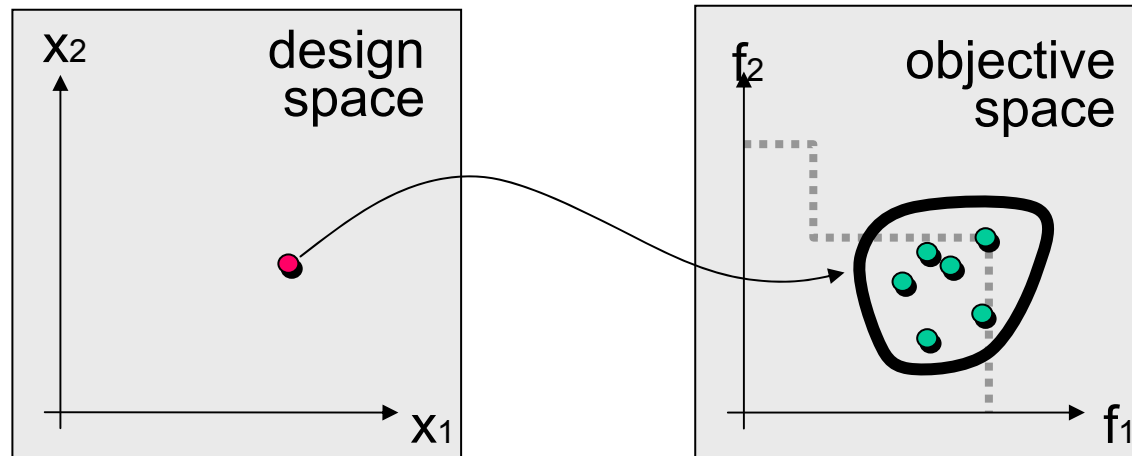
## How to...

- handle multiple objectives
- account for uncertainty
- optimize for robustness
- incorporate user preferences
- achieve diversity in the parameter space
- reduce the design space complexity
- statistically assess the quality of the outcome
- simplify implementation and maximize reusability
- deal with limited memory resources
- postprocessing of tradeoff surface

# Uncertainty in Multiobjective Optimization

**Uncertainty** = each time a solution/design is evaluated, possibly different objective function values emerge due to

- stochastic system model (Monte Carlo simulation)
- model parameter variations (cost estimates)



**Previous work:** [Hughes:01;Teich:01; Goldberg et al.:03,05]

- assume a certain type of distribution (uniform,normal)
- mixed models (stochastic dominance, regular distance)

# Indicators and Uncertainty

## Deterministic model:

$$\operatorname{argmin}_{S \in \mathcal{M}(X)} I(f(S), f(R))$$

each solution is associated with one objective vector

## Stochastic model:

$$\operatorname{argmin}_{S \in \mathcal{M}(X)} E(I(\mathcal{F}(S), \mathcal{F}(R)))$$

each solution is associated with a random variable

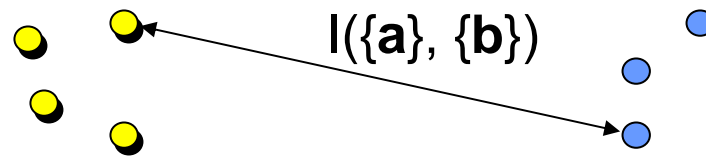
$$\begin{aligned} E(I(\mathcal{F}(S), \mathcal{F}(R))) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\mathcal{F}(S) = A, \mathcal{F}(R) = B) \cdot I(A, B) \, dA dB \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\mathcal{F}(S) = A) \cdot P(\mathcal{F}(R) = B) \cdot I(A, B) \, dA dB \end{aligned}$$

[Basseur, Zitzler:05]

# Estimating the Expected $\varepsilon$ -Value

$S(\mathbf{x})$  = sample of objective vectors for solution  $\mathbf{x}$

**Main idea:** consider all combinations of objective vectors and compute expected (mean) indicator value



$$\hat{E}(I(\mathcal{F}(S), \{\mathbf{z}^*\})) = \sum_{\mathbf{z}_1 \in \mathcal{S}(\mathbf{x}_1)} \sum_{\mathbf{z}_2 \in \mathcal{S}(\mathbf{x}_2)} \cdots \sum_{\mathbf{z}_m \in \mathcal{S}(\mathbf{x}_m)} \frac{I(\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m\}, \{\mathbf{z}^*\})}{\prod_{1 \leq i \leq m} |\mathcal{S}(\mathbf{x}_i)|}$$

$$\hat{E}(I(\mathcal{F}(S), \mathcal{F}(\{\mathbf{x}^*\}))) = \frac{1}{|\mathcal{S}(\mathbf{x}^*)|} \sum_{\mathbf{z}^* \in \mathcal{S}(\mathbf{x}^*)} \hat{E}(I(\mathcal{F}(S), \{\mathbf{z}^*\}))$$

**Alternative:** mean value per objective function  
(loses distribution characteristics)



# Preliminary Simulation Results

indicator-based

average [Hug01]

		EIV		BCK		Exp		Avg		PDR	
		Z	X	Z	X	Z	X	Z	X	Z	X
n=1	EIV			> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%
	BCK	> 5%	> 5%			> 5%	> 5%	> 5%	> 5%	> 5%	> 5%
	Exp	> 5%	> 5%	> 5%	> 5%			> 5%	> 5%	> 5%	> 5%
	Avg	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%			> 5%	> 5%
	PDR	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%		
n=5	EIV			> 5%	> 5%	> 5%	1.56%	0.014%	0.014%	0.014%	0.014%
	BCK	> 5%	> 5%			1.89%	0.041%	0.014%	0.014%	0.014%	0.014%
	Exp	> 5%	> 5%	> 5%	> 5%			0.014%	0.014%	0.014%	0.014%
	Avg	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%			2.28%	> 5%
	PDR	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	0.019%		
n=10	EIV			> 5%	> 5%	> 5%	> 5%	0.014%	0.014%	0.014%	0.014%
	BCK	> 5%	> 5%			> 5%	0.45%	0.014%	0.014%	0.014%	0.014%
	Exp	> 5%	> 5%	> 5%	> 5%			0.014%	0.014%	0.014%	0.014%
	Avg	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%			> 5%	> 5%
	PDR	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	1.28%	> 5%		

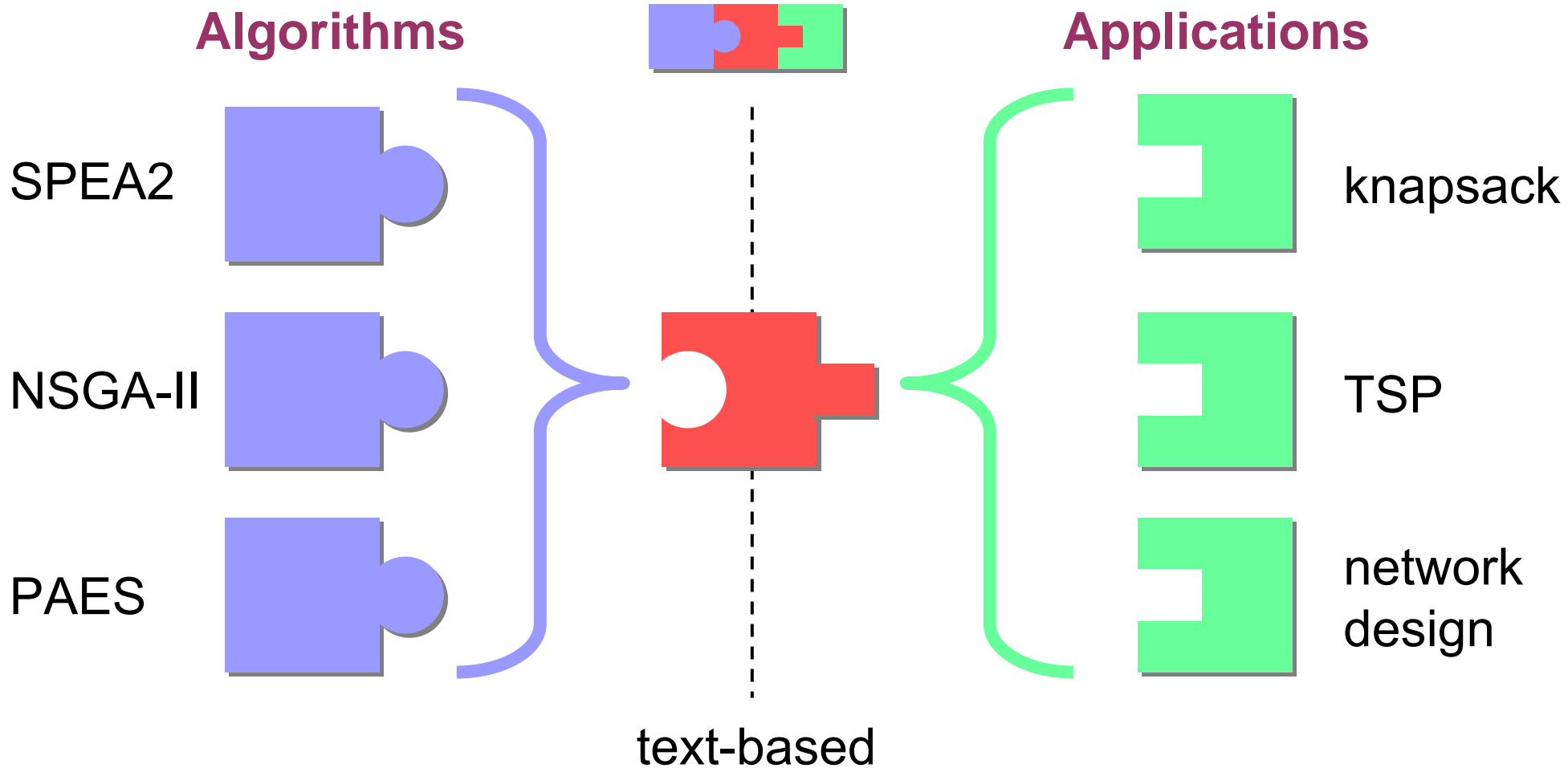
- no significant differences in <3 objectives
- highly significant differences in higher dimensions

# Key Issues in Design Space Exploration Using RSA

## How to...

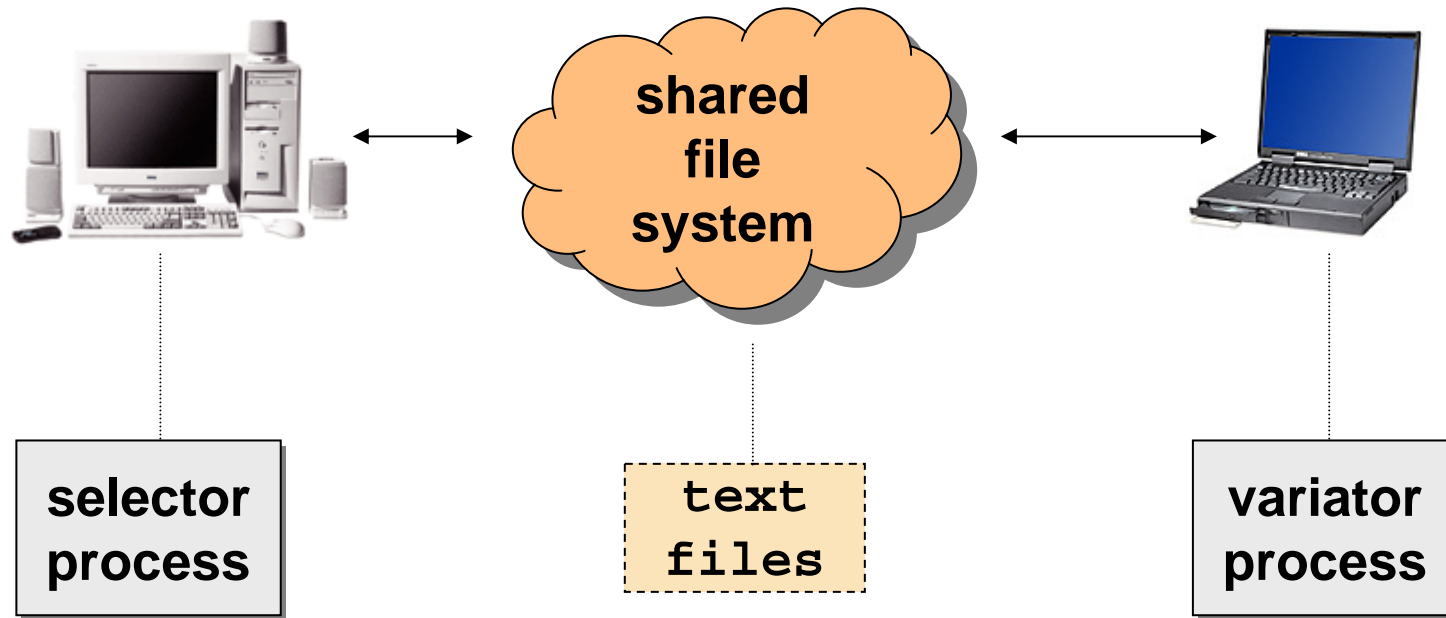
- handle multiple objectives
- account for uncertainty
- optimize for robustness
- incorporate user preferences
- achieve diversity in the parameter space
- reduce the design space complexity
- statistically assess the quality of the outcome
- simplify implementation and maximize reusability
- deal with limited memory resources
- postprocessing of tradeoff surface

# The Concept of PISA



**Platform and programming language independent Interface  
for Search Algorithms** [Bleuler et al.:03; Künzli et al. 05]

# 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



ETH Zürich > IT & E

## PISA

### A Platform

PISA is a text-based interface for evolutionary search algorithms. It splits an optimization process into two separate programs. One implementing the search algorithm, the other implementing the parts of the search which are independent of the problem to be solved.

### Contents

[About PISA](#)

[For beginners](#)

[↓ News](#)

[↓ Available modules](#)

[↓ Relevant Publications](#)

[Licensing](#)

[Specification \(pdf\)](#)

[Bugs](#)

[How to write a module?](#)

[How to submit](#)

[People and contacts](#)

**New:** A new version of the DTLZ module is available, it fixes a bug in the ZDT3 test function. See [bugs](#) for more details.



## Optimization Problems (variator)

### LOTZ - Demonstration Program ([more...](#))

- Source: in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### LOTZ2 - Leading Ones Trailing Zeros ([more...](#))

- Source: in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### Knapsack Problem ([more...](#))

- Source: in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### EXPO - N

- Binaries: (incl. JRE 1.4.1) [Solaris](#), [Windows](#), [Linux](#)
- Binaries: (pure JAVA, no JRE) [All platforms](#)

### DTLZ - Continuous Test Functions ([more...](#))

- Source: in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### BBV - Biobjective Binary Value Problem ([more...](#))

- Source: in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### MLOTZ - Generalization of the LOTZ Problem ([more...](#))

- Source: in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)



## Optimization Algorithms (selector)

### SEMO - Demonstration Program ([more...](#))

- Source in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### SEMO2 - Simple Evolutionary Multiobjective Optimizer ([more...](#))

- Source in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### FEEMO - Fast Evolutionary Multiobjective Optimizer ([more...](#))

- Source in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### SPEA2 - Strength Pareto Evolutionary Algorithm 2 ([more...](#))

- Source in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### NSGA2 - Nondominated Sorting Genetic Algorithm 2 ([more...](#))

- Source in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### ECEA - Epsilon-Constraint Evolutionary Algorithm ([more...](#))

- Source in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

### IBEA - Indicator Based Evolutionary Algorithm ([more...](#))

- Source in [C](#)
- Binaries: [Solaris](#), [Windows](#), [Linux](#)

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

# Network Processor Design Application (EXPO)

The screenshot displays the EXPO software interface, which is used for design space exploration. It consists of several windows and panels:

- Control Panel:** Features buttons for 'Run/Pause' and 'Reset'. A 'stop after generation' field is set to 15000. A text area shows the progress of the simulation, including 'Variation finished.', '\*\*\*\*\* Generation 3 \*\*\*\*\*', and 'All active gene IDs read.'.
- Current Population Plot:** A scatter plot titled 'current population' showing the distribution of the current population. The x-axis is labeled 'x axis' and ranges from -1.4 to -0.0 (scaled by  $\times 10^2$ ). The y-axis is labeled 'y axis' and ranges from 0.1 to 0.6. Red dots represent individual elements in the population.
- Implementation Details:** A window titled 'Implementation Nr. 0' shows options to 'Save SVG', 'Save JPG', and 'Save PNG'. It also includes 'close' and 'Scenarios: RT, NRT' buttons.
- Hardware Utilization:** A diagram showing the utilization of various hardware components: ARM9 (67%), DSP (1%), LookUp (0%), CheckSum (0%), and PowerPC (36%). Below this, a large double-headed arrow indicates the flow of data through the system.
- Flow and Queue Information:** At the bottom, it shows 'Flow: RTSend', 'Priority: 3', and 'Acc. Waiting Time in Queue: 9.536'. Below these are labels for 'Schedule', 'RTPtx', 'UDPTx', 'ARPLU', 'BuildIP', 'LinkTx', and 'RouteLU1'.

[Thiele, Künzli et al.:03,04,05]

# Open Issues in Design Space Exploration Using RSA

## How to...

- handle multiple objectives
- account for uncertainty
- optimize for robustness
- incorporate user preferences
- achieve diversity in the parameter space
- reduce the design space complexity
- statistically assess the quality of the outcome
- simplify implementation and maximize reusability
- deal with limited memory resources
- postprocessing of tradeoff surface

# Acknowledgements

- Matthieu Basseur, University of Lille, France
- Stefan Bleuler, ETH Zurich, Switzerland
- Simon Künzli, ETH Zurich, Switzerland
- Lothar Thiele, ETH Zurich, Switzerland