Evolutionary Computation for Modelling and Optimization in Finance

Sandra Paterlini

CEFIN & RECent, University of Modena and Reggio E., Italy

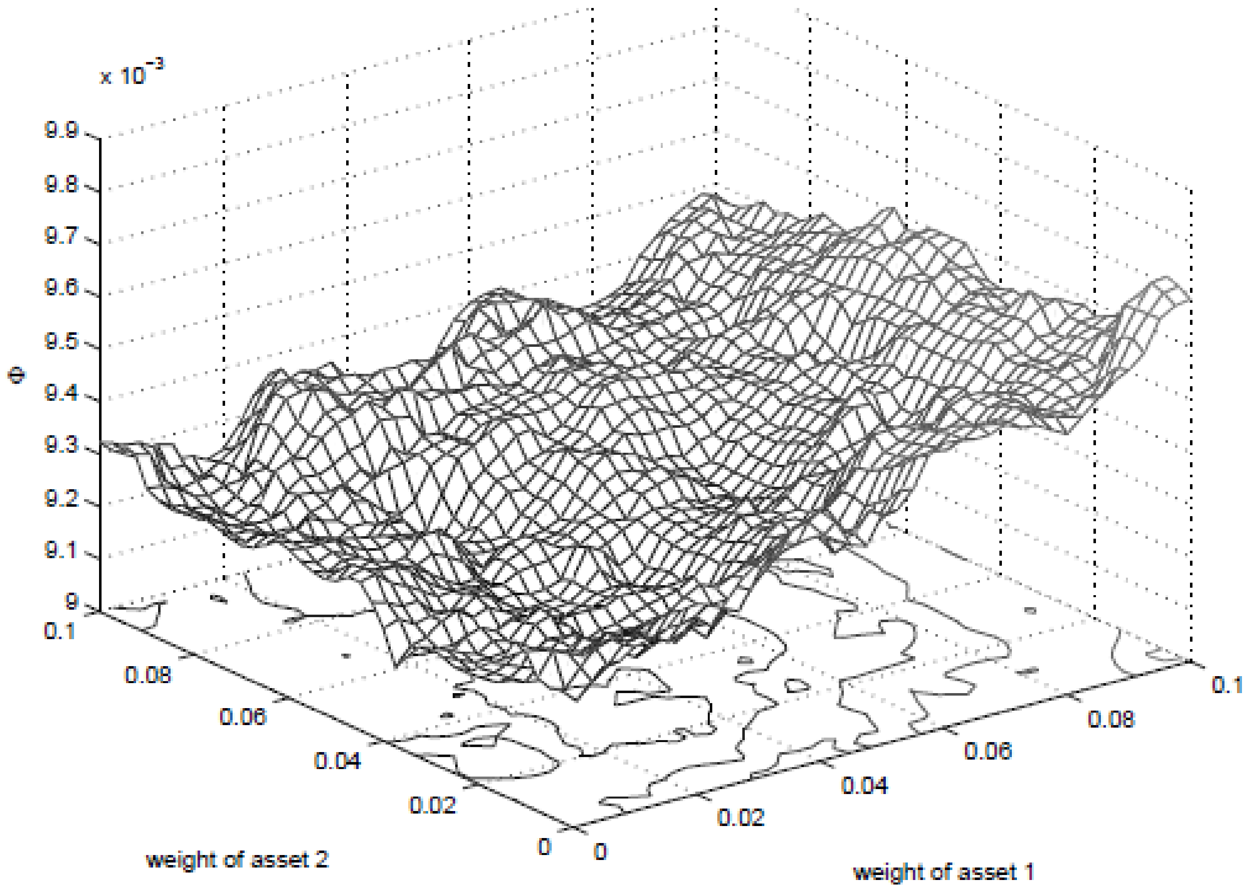
– Introduction

Why do we need Evolutionary Computation (EC)?

- conventional techniques require rigid assumptions (convexity, linearity, differentiability, explicitly defined objectives, problem can be split into subproblems, etc.)
- often the objective function is discontinuous, multimodal, has plateaus, etc.
- many discrete real-world problems are computational hard, i.e., an exact solution cannot be computed in reasonable time
- often we get away with simplifications (linearization, convexication, etc.), but not in all cases!
- conventional techniques lack generality; new problem solutions require new implementation
- …but EAs are not always necessary!

– Why do we need EC in finance?

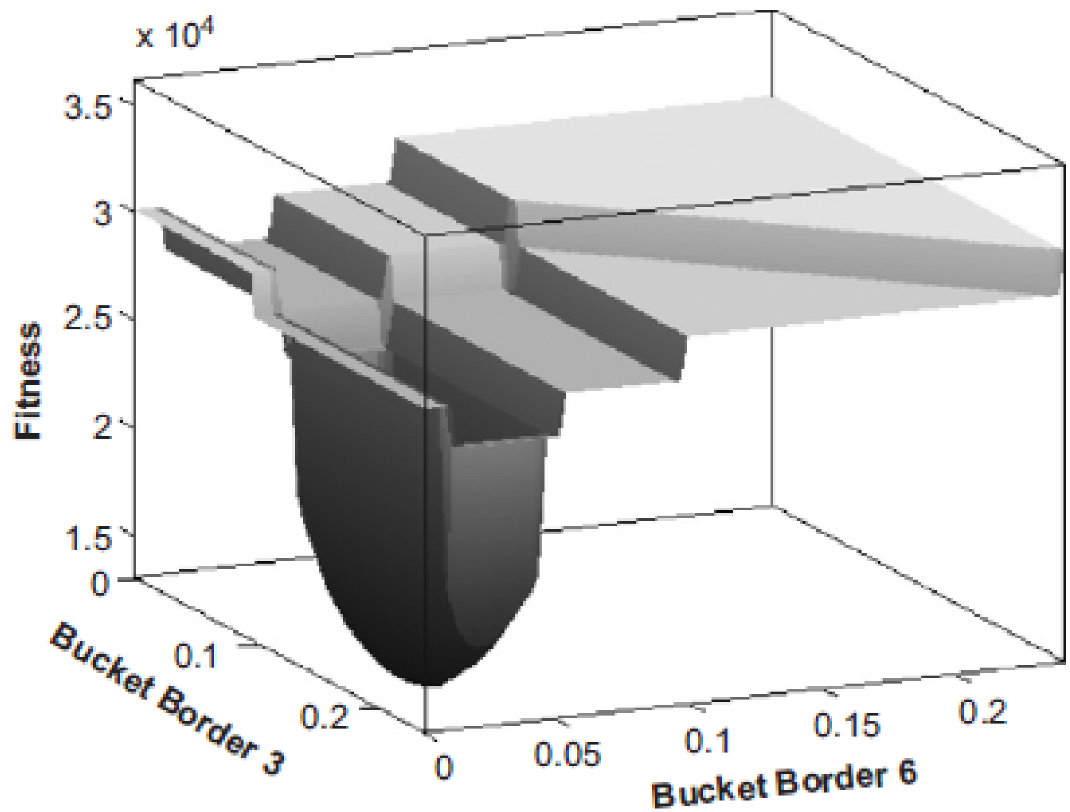
Multimodality: Value-at-Risk Optimization



Gilli and Schumann, 2008

– Why do we need EC in finance?

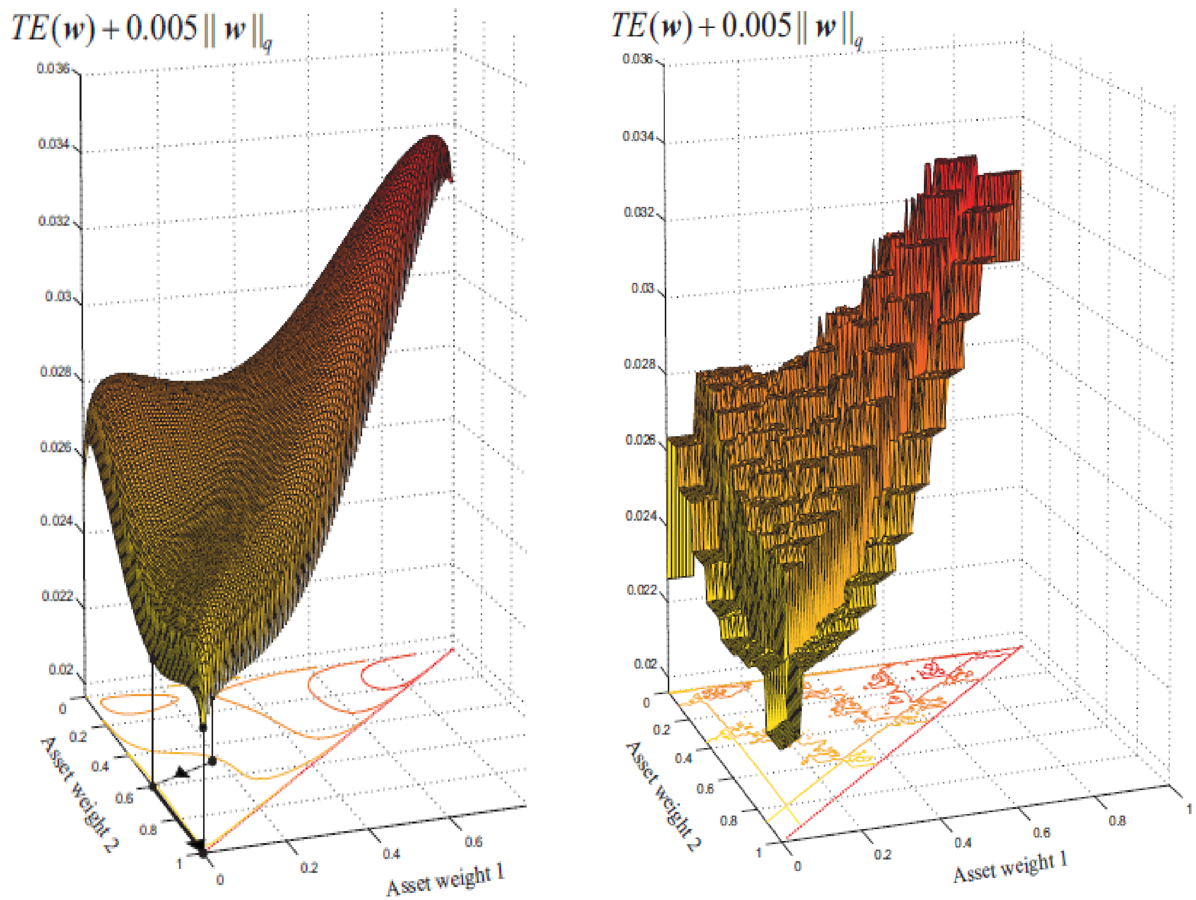
Plateaus: Credit Risk Bucketing



Krink, Paterlini and Resti 2008

– Why do we need EC in finance?

Constraints: Index Tracking using q-norm penalty



Fastrich, Paterlini and Winker, 2010

– EAs and related search heuristics

Evolutionary algorithms (EA)

GA GP EP ES DE

- GA: Genetic Algorithms
- GP: Genetic Programming
- EP: Evolutionary Programming
- ES: Evolution Strategies
- DE: Differential evolution

Related heuristics

SA GLS TS AS PS

- SA: Simulated Annealing, Hill Climbing
- GLS: Guided local search
- TS: Tabu search
- AS: Ant systems
- PS: Particle swarms

– EAs and related search heuristics

Common denominator in EAs

- Algorithms that deal with a population of individuals, which are selected and altered in an iterative process.

Why are EAs different from local search?

- EAs and related heuristics are based on the idea of competition and recombination of candidate solutions in a population

– Pseudo-code

```
void EvolutionaryAlgorithm()
{
    t = 0;
    initialise population P(t);   // create random solutions
    evaluate population P(t);   // calculate fitnesses
    while (not termination condition) {
        t = t + 1;
        select next generation P(t) from P(t-1);
        alter P(t);   // mutate and recombine genes
        evaluate population P(t);   // calculate fitnesses
    }
}
```

P(t) = Population at time t

Multiobjective
Financial Portfolio Optimization

– Multiobjective Portfolio Optimization

Mean-VaR$_{(1-\alpha)}$ Selection

$$(1a) \qquad \min \quad f_1(\mathbf{w}') = \overline{L} - VaR_{1-\alpha}$$

$$(1b) \qquad \max \quad f_2(\mathbf{w}') = \overline{L}$$

subject to:

$$(1c) \quad \sum_{i=1}^{N} w_i = 1 \quad 0 \leq w_i \leq 1$$

$(1d)$ – ... other non linear constraints

where

$\overline{L} = \sum_{i=1}^{n_s} L_s \Big/ n_s$ corresponds to the expected absolute return of the portfolio

Main challenges

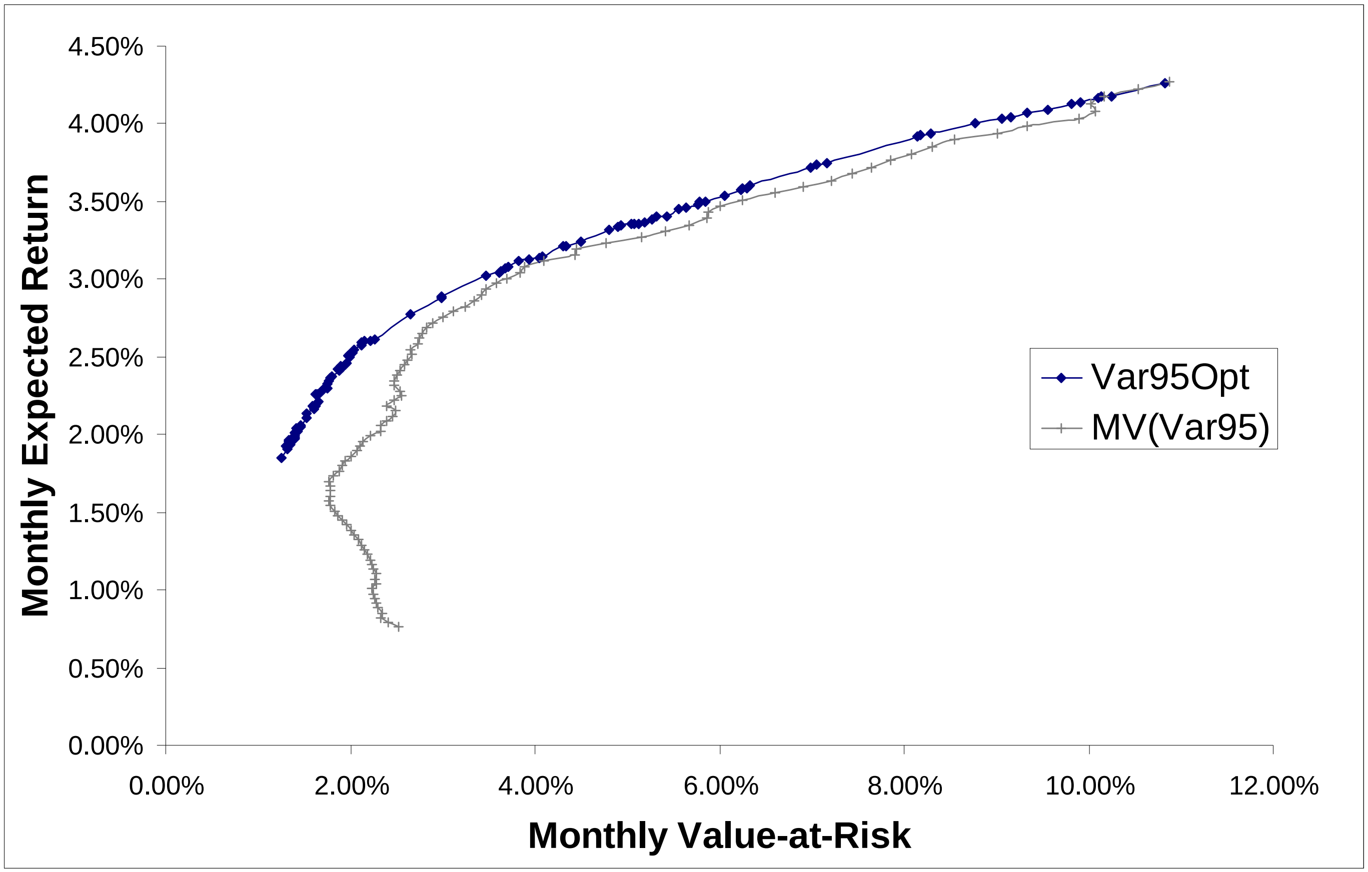
- •Non-linear objective function
- •Multiobjective Problem
- •Non-linear constraints

Idea

Evolve a set of Pareto optimal solutions, which are represented by the population of the evolutionary algorithm

– Multiobjective Portfolio Optimization

Mean-VaR$_{(1-\alpha)}$ Selection



– EAs in Multiobjective Optimization

Domination and Diversity preservation

- During selection between two individuals *i* and *j* select the one which is **dominated** or **constrained-dominated** by less other solutions
- if they are dominated by the same number of solutions then select the one with the **larger distance** value (i.e. crowding operator)

– EAs in Multiobjective Optimization

Dominance

Let F be a multiobjective minimization problem with p objectives f_k with $k = 1,\ldots, p$

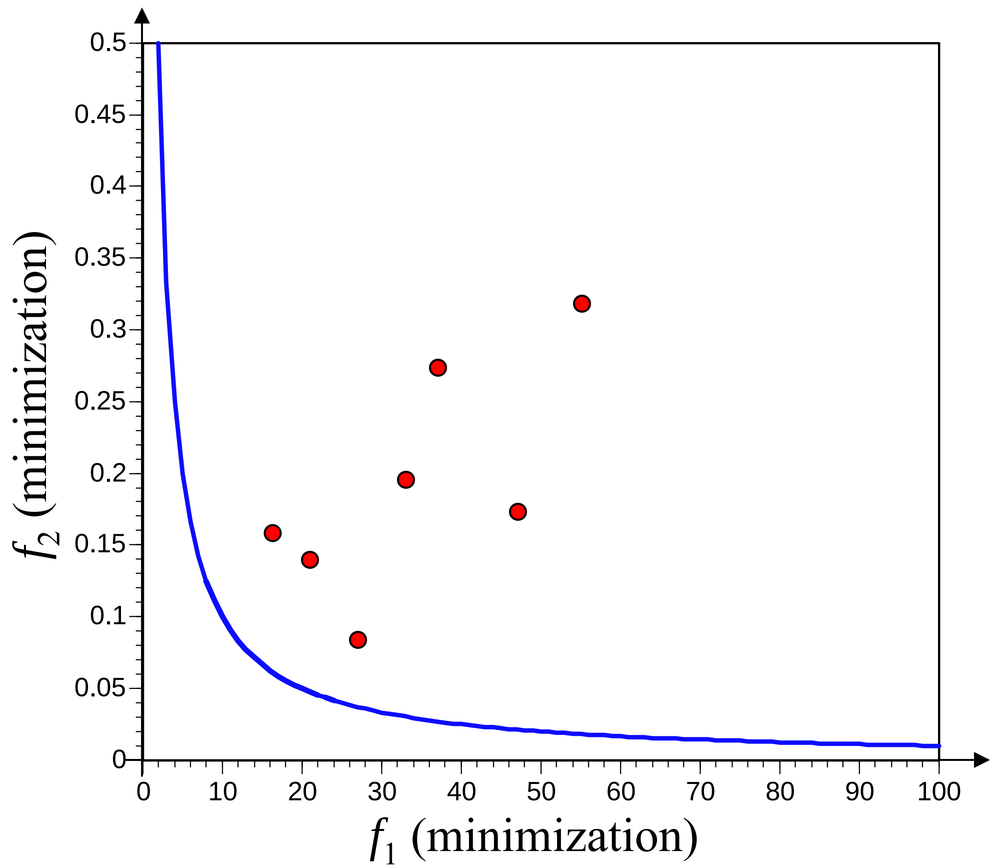
A candidate solution $\mathbf{x}_i$ **dominates** $\mathbf{x}_j$ if and only if

$$\forall\, k,\ \ f_k(\mathbf{x}_i) \leq f_k(\mathbf{x}_j)$$

$$\exists\, l,\ f_l(\mathbf{x}_i) < f_l(\mathbf{x}_j)$$

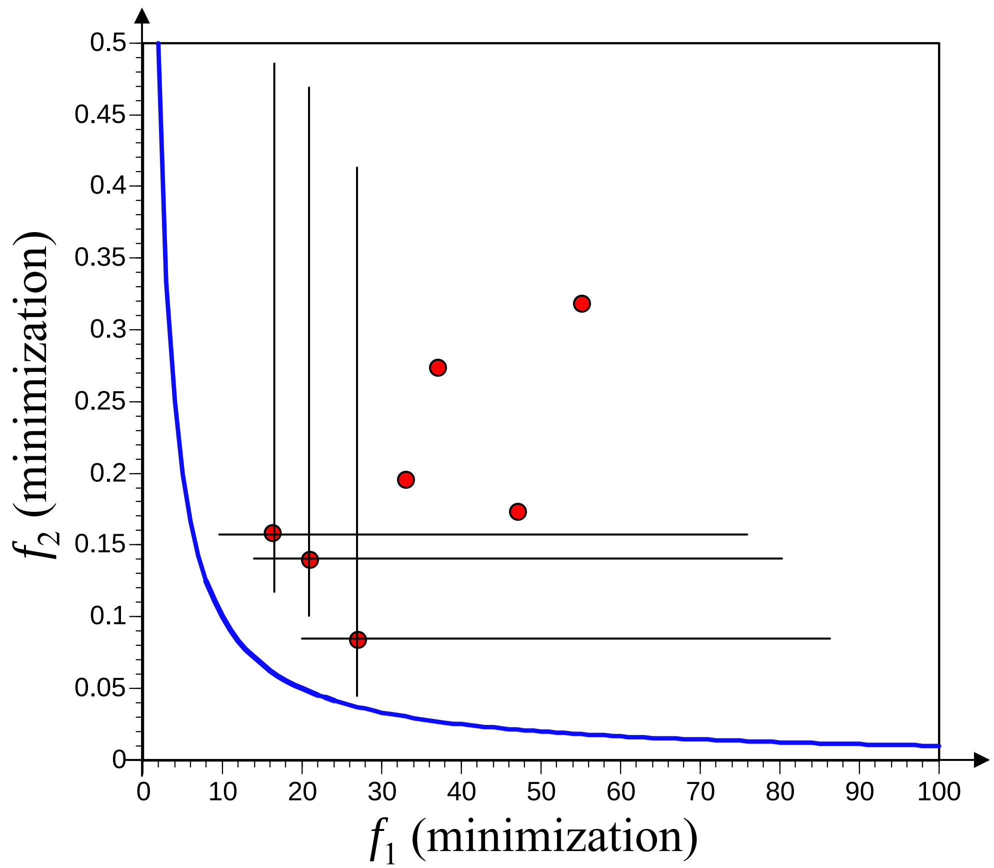
– EAs in Multiobjective Optimization

Determination of the non-dominated fronts



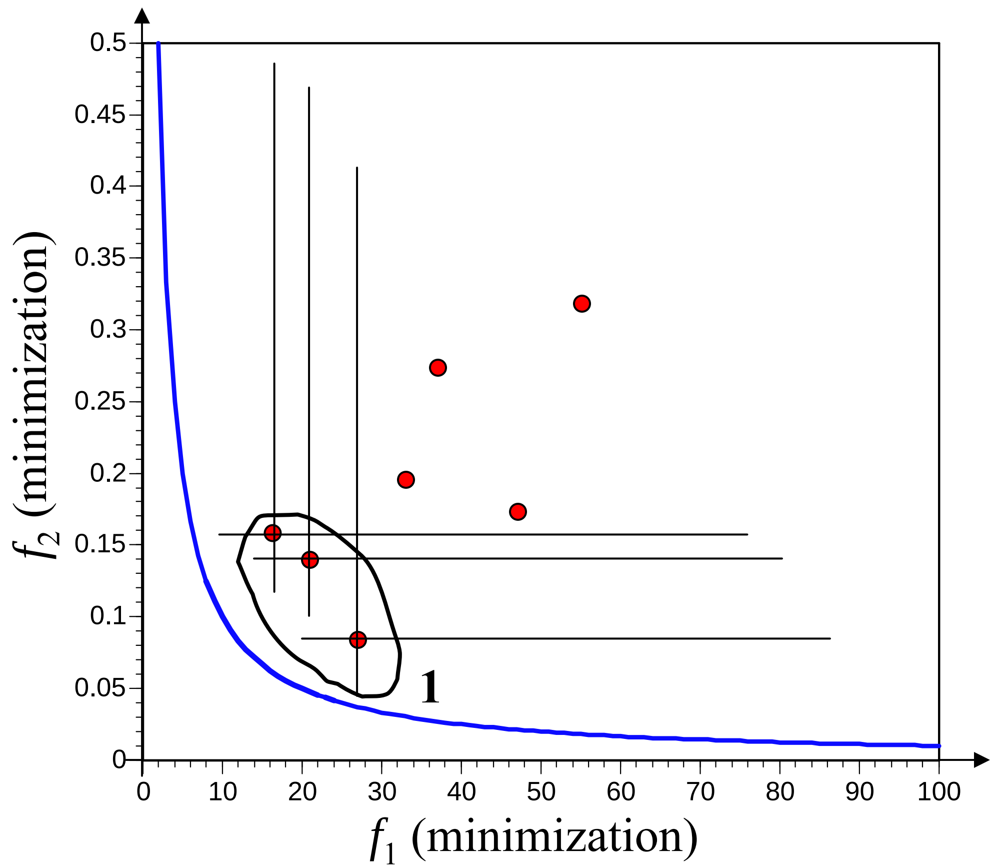
– EAs in Multiobjective Optimization

Determination of the non-dominated fronts



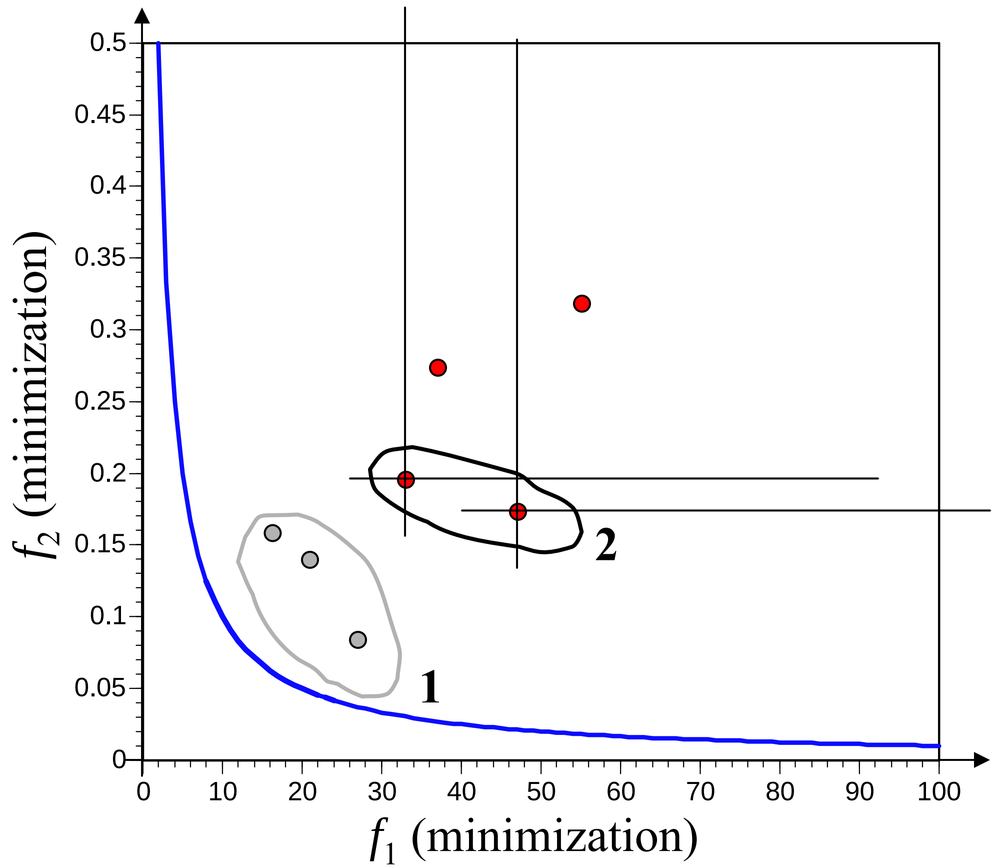
– EAs in Multiobjective Optimization

Determination of the non-dominated fronts



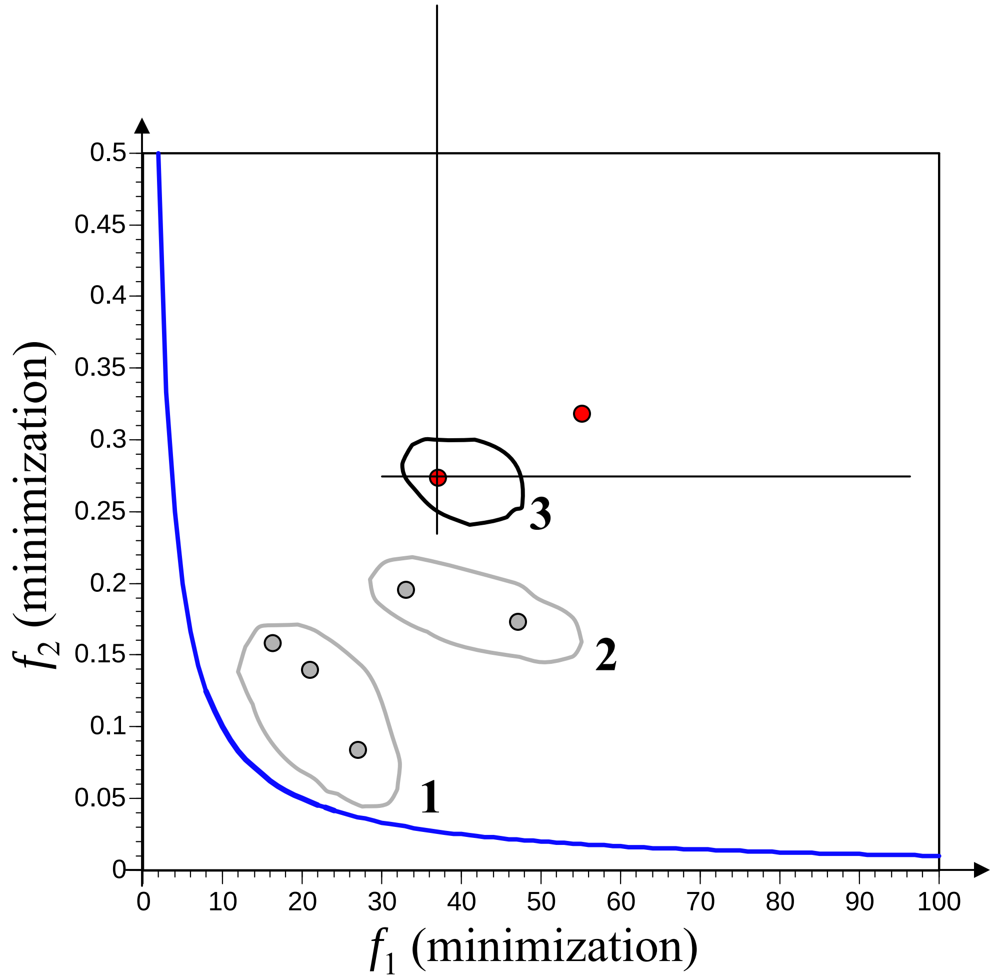
– EAs in Multiobjective Optimization

Determination of the non-dominated fronts



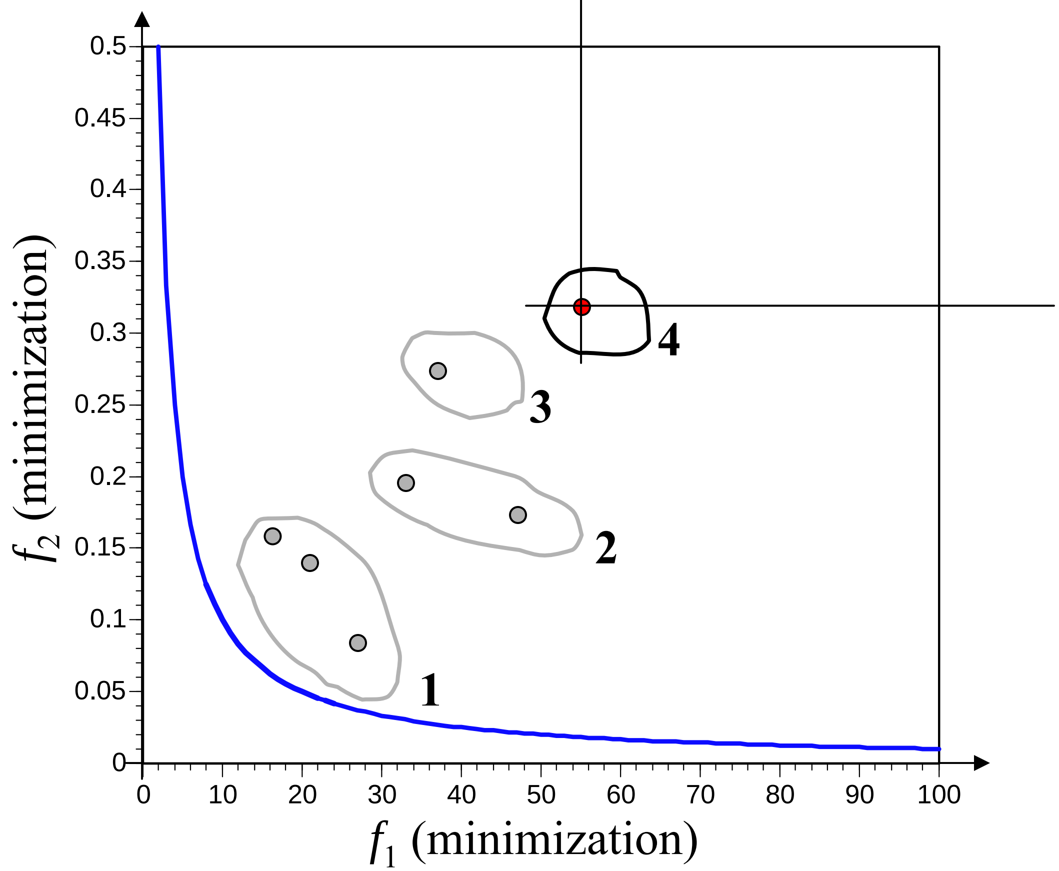
– EAs in Multiobjective Optimization

Determination of the non-dominated fronts



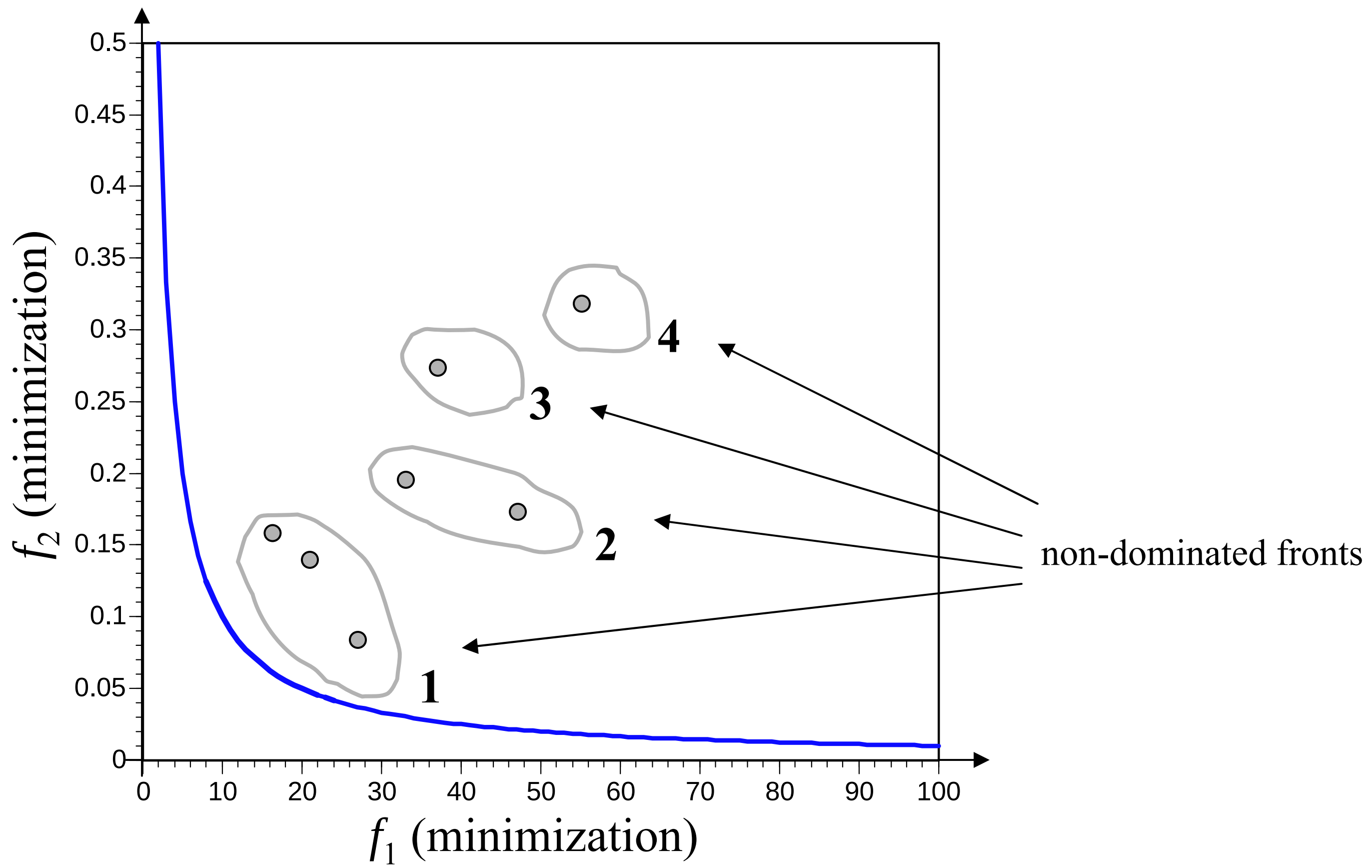
– EAs in Multiobjective Optimization

Determination of the non-dominated fronts



– EAs in Multiobjective Optimization

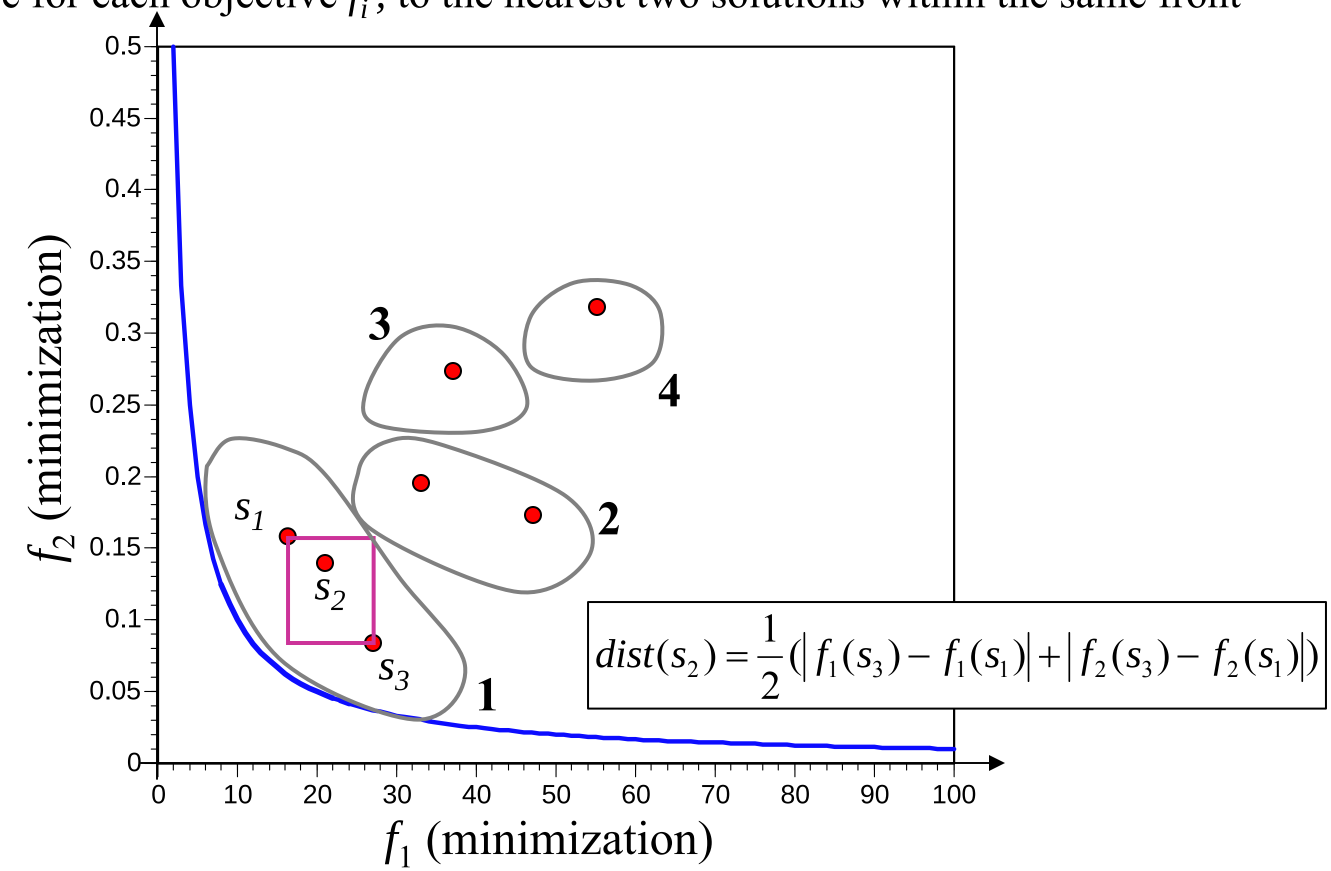
Determination of the non-dominated fronts



– EAs in Multiobjective Optimization

Diversity preservation

for each solution calculate the cuboid distance, i.e., the mean of the lower and upper distance for each objective f_i, to the nearest two solutions within the same front

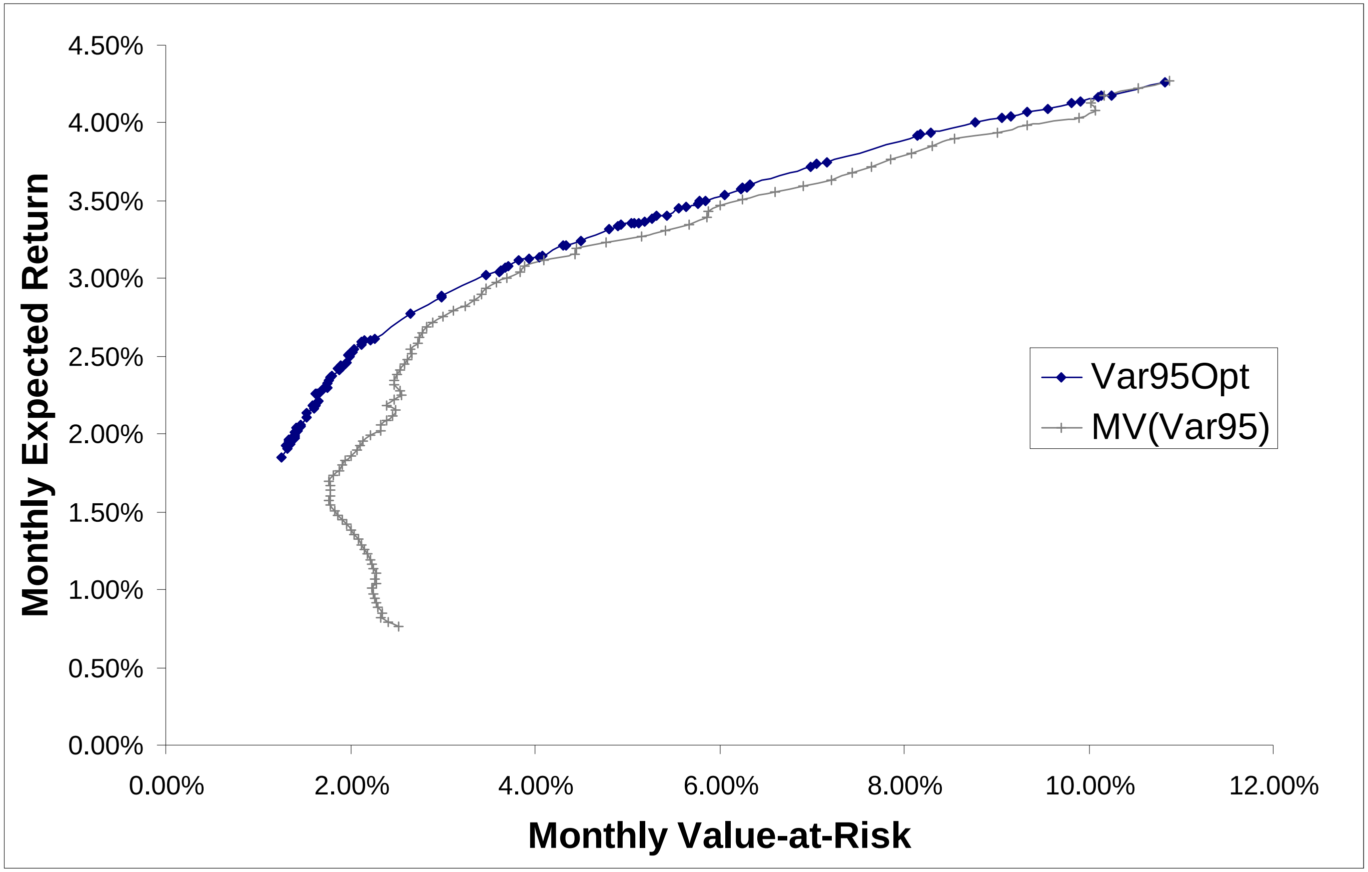


– EAs in Multiobjective Optimization

```
void MultiObjectiveEvolutionaryAlgorithm()
{
 t=0;
 Initialize population P(t); // create random solutions
 Evaluate population P(t); // calculate fitnesses
 while (not termination condition){
          t=t+1;
          Determination of the constrained non-dominated fronts {
              Calculate non-domination ranks
              Diversity preservation }
          Multi-objective fitness assignment
          Select next generation P(t) from P(t-1);
          Alter P(t); // Mutate and recombine genes
          Evaluate population P(t); // calculate fitnesses  }
}
```

– Multiobjective Portfolio Optimization

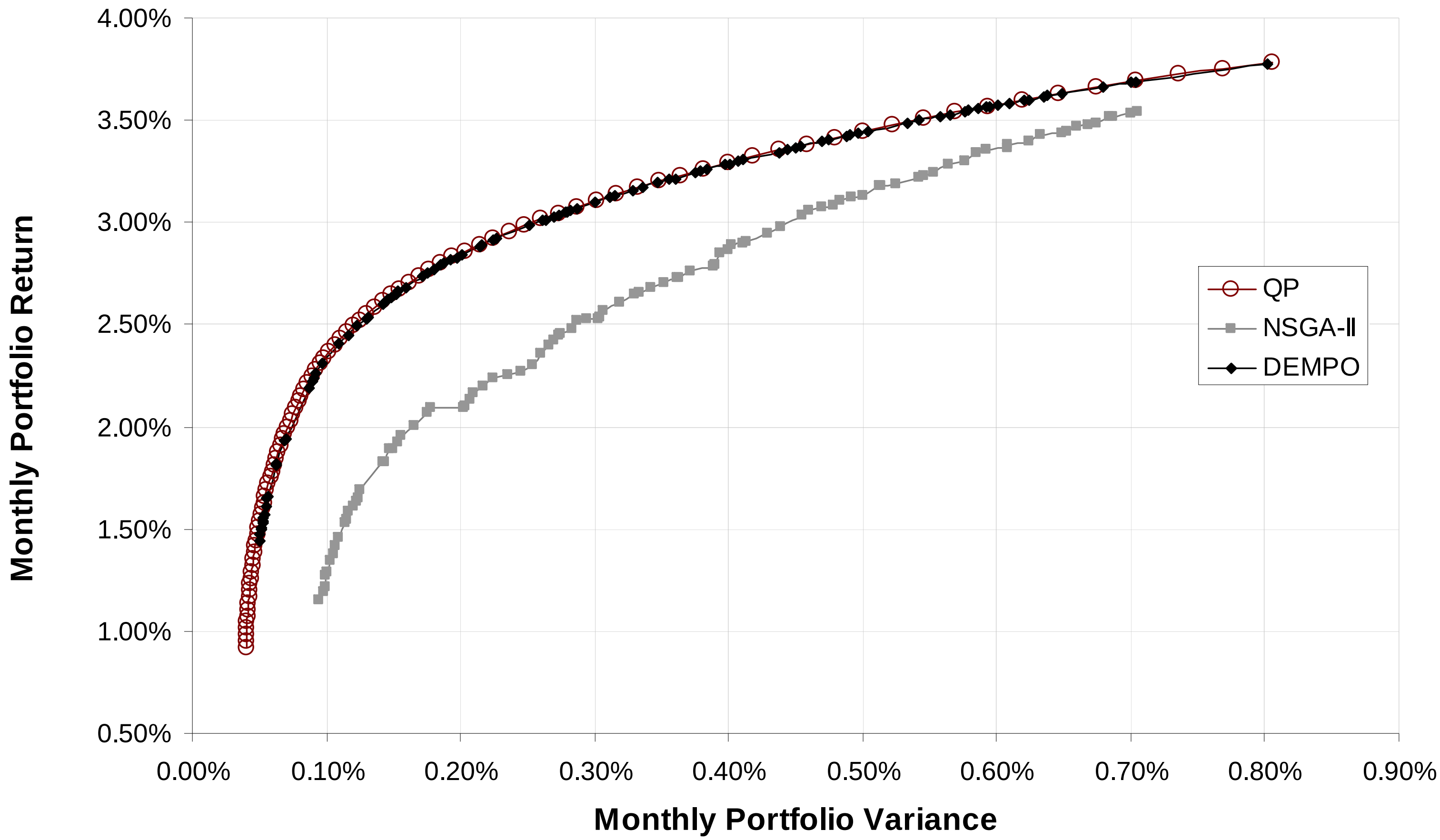
Mean-VaR$_{(1-\alpha)}$ Selection



Krink and Paterlini, 2009

– Multiobjective Portfolio Optimization

Main Challenges



Krink and Paterlini, 2009

– EAs in Multiobjective Optimization

Main challenges

- Multiobjective problems: use the whole population to determine the Pareto front
- Diversity preservation
  - Sharing versus Crowding
  - Distance Measures
- Constraint-handling
- Parameter Tuning
- Statistical Analysis of Convergence Properties
- Performance criteria in MO
- …

– EC in Finance

Conclusions

- Hardware development, data availability and speed up methods support on-going research
- Financial industry shows a growing interest towards heuristic optimization methods
  - Financial Portfolio Selection
  - Option Pricing
  - Model Calibration
  - ...
- Many financial problems can be effectively tackled by EC, as shown by many published papers and books

Conclusions

- Many more will be published...

Computational Statistics & Data Analysis

3rd Special Issue on

OPTIMIZATION HEURISTICS IN ESTIMATION AND MODELLING PROBLEMS

Submission deadline: **January 15, 2011**

– References

References

- Gilli, M. and E. Schumann, Heuristic Optimization in Financial Modelling, WP
- Lyra, M., J. Paha, S. Paterlini and P. Winker, in press, Optimization Heuristics for Determining Internal Rating Grading Scales, Computational Statistics & Data Analysis, doi:10.1016/j.csda.2009.03.004.
- Krink, T. and S. Paterlini, in press, Multiobjective Optimization using Differential Evolution for Real-World Portfolio Optimization, Computational Management, DOI 10.1007/s10287-009-0107-6.
- Krink, T., S. Mittnik and S. Paterlini, 2009, Differential Evolution and Combinatorial Search for Constrained Index Tracking, Annals of Operation Research, 172, 153-176.
- Krink, T., S. Paterlini and A. Resti, 2008, The optimal structure of PD buckets, Journal of Banking and Finance, 32, 10, 2275-2286.
- Krink, T., S. Paterlini and A. Resti, 2007, Using Differential Evolution to improve the accuracy of bank rating systems, Computational Statistics & Data Analysis, Elsevier, 52/1, 68-87.