Evolutionary Multiobjective Optimisation and Uncertainty

Juergen Branke

We live in an uncertain world

- Uncertain customer demand
- Uncertain travel times
- Volatile stock market
- Manufacturing tolerances
- …




Two forms of uncertainty

- Aleatoric uncertainty: statistical uncertainty, nothing an experimenter can do about
- Epistemic uncertainty: due to things we could in principle know, but in practice don't. Can be reduced by gathering more data or refining models

Outline

- Uncertainty about preferences
- Uncertainty as additional objective
- Optimising noisy objectives
  - expected performance
  - worst case performance
- Summary

Multi-objective Optimization =
Single-objective optimization
+ uncertainty about user preferences

Perfect knowledge of preferences

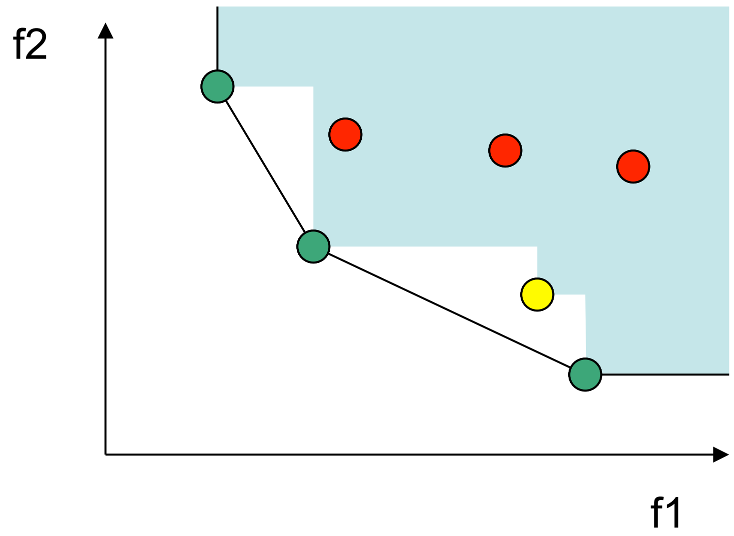
- would allow us to rank all solutions
- problem would effectively be a single objective problem

Uncertain user preferences

- They have not yet been expressed
- They are difficult to express in closed form
- User has not formed an opinion yet
- User is inconsistent
- There may be multiple users

Different degrees of uncertainty

- No knowledge
- Monotonic
  -> Pareto dominance
- Linear
- Probability distribution



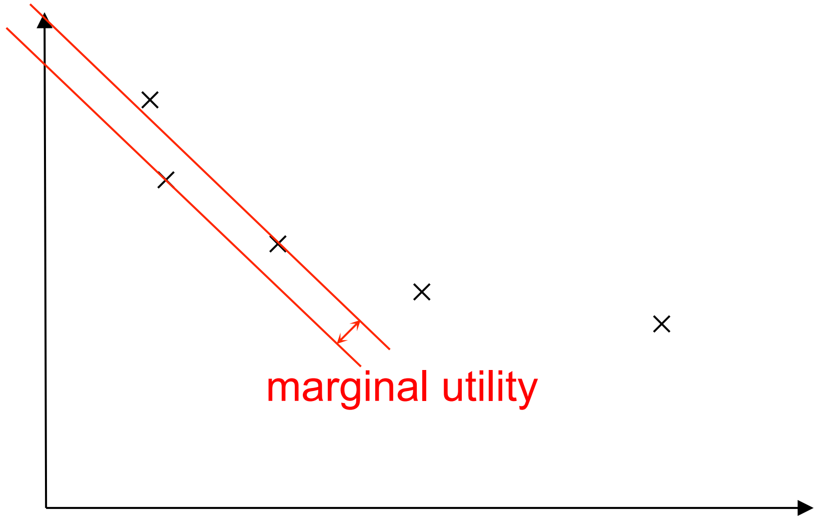
Evaluating Pareto fronts

- A probability distribution over utility functions allows us to quantify the quality of a solution set: expected utility of the chosen solution

$$E(U(X)) = \int_{u \in U} P(u) \max_{x \in X} u(x) du$$

Marginal Utilities [Branke, Deb, Dierolf, Oswald 2004]

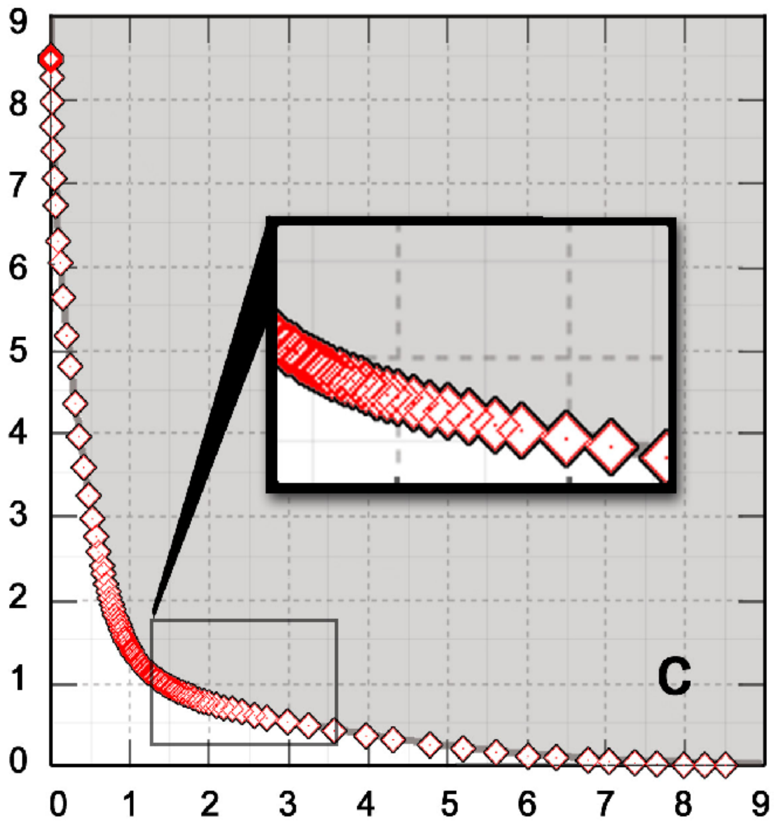
- Assume linear utility function
- Evaluate each solution with expected loss of utility if solution would not be there

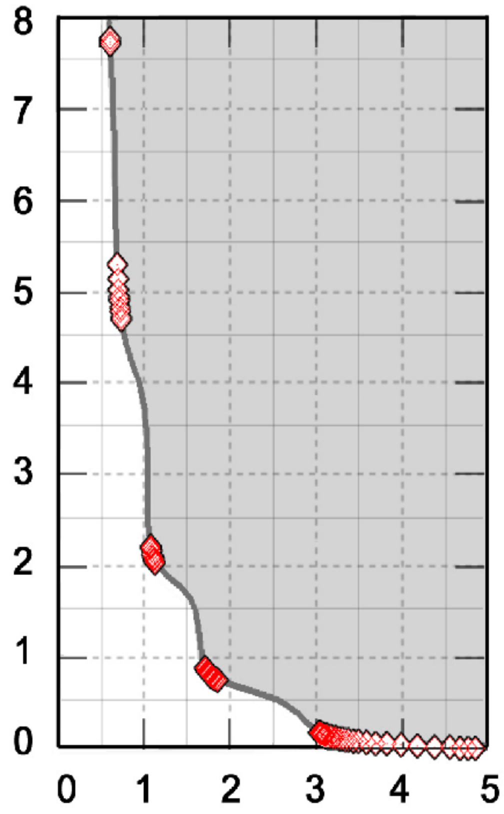


$$E(U'(x_i)) = \int_{\alpha=\lambda_{i-1,i+1}}^{\lambda_{i-1,i}} \alpha(f_1(x_i) - f_1(x_{i-1})) + (1-\alpha)(f_2(x_i) - f_2(x_{i-1}))d\alpha$$
$$+ \int_{\alpha=\lambda_{i,i+1}}^{\lambda_{i-1,i+1}} \alpha(f_1(x_i) - f_1(x_{i+1})) + (1-\alpha)(f_2(x_i) - f_2(x_{i+1}))d\alpha$$

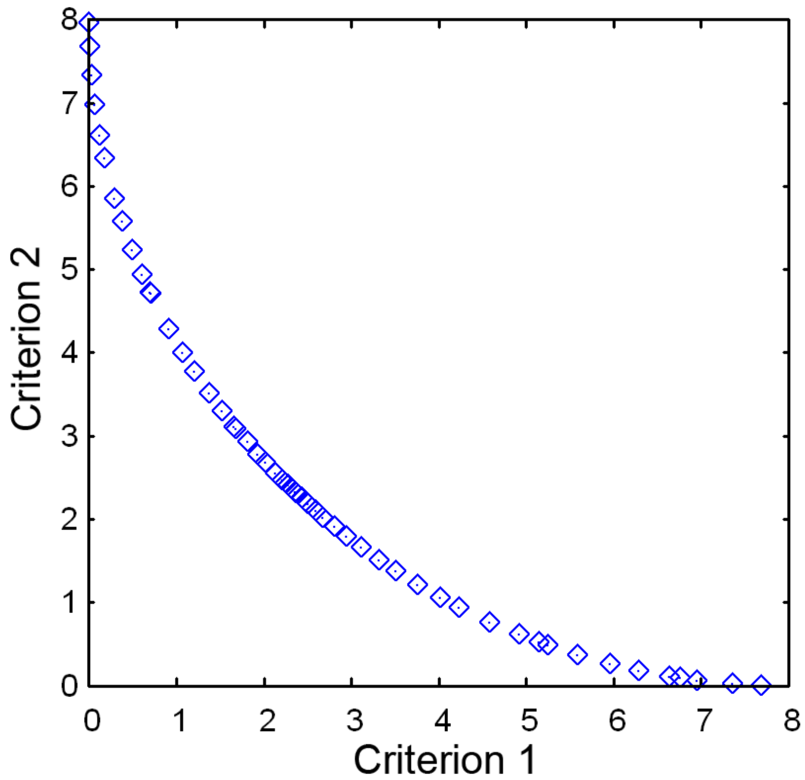
Finding knees [Branke, Deb, Dierolf, Oswald 2004]

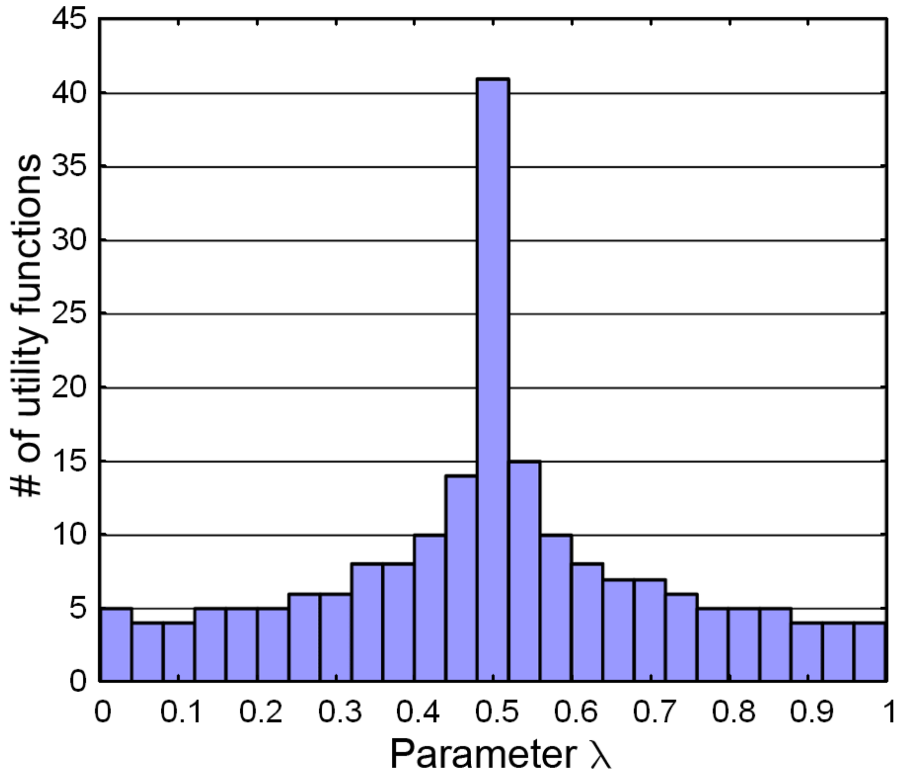
Solution where a small improvement in either objective will lead to a large deterioration in the other





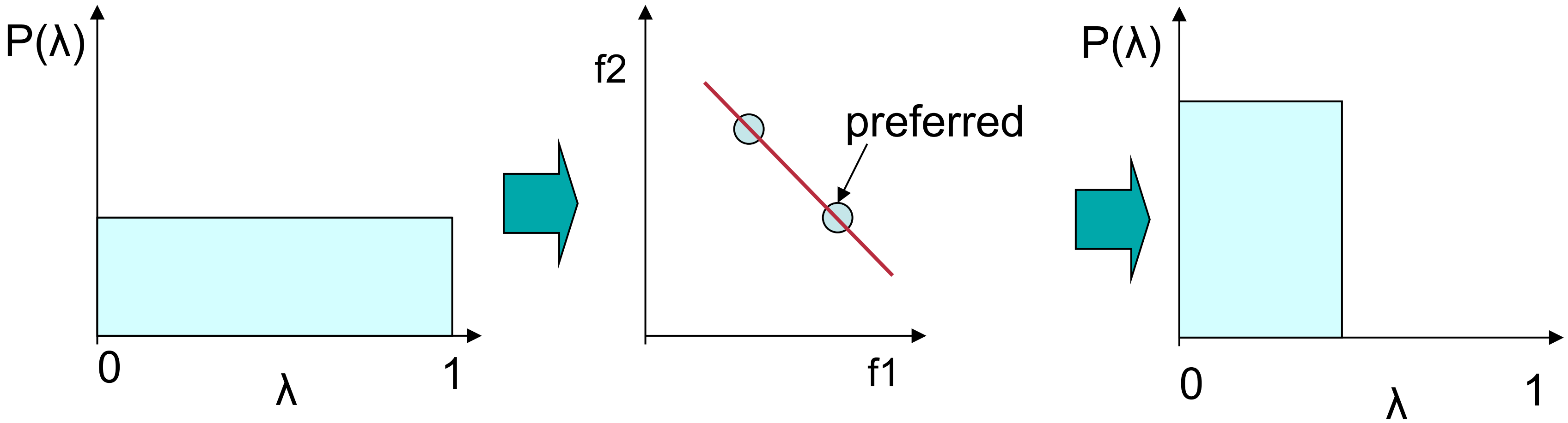
Non-uniform distributions [Branke 2008]





Learning preferences

- User's rankings of pairs of solutions may restrict the space of compatible utility functions

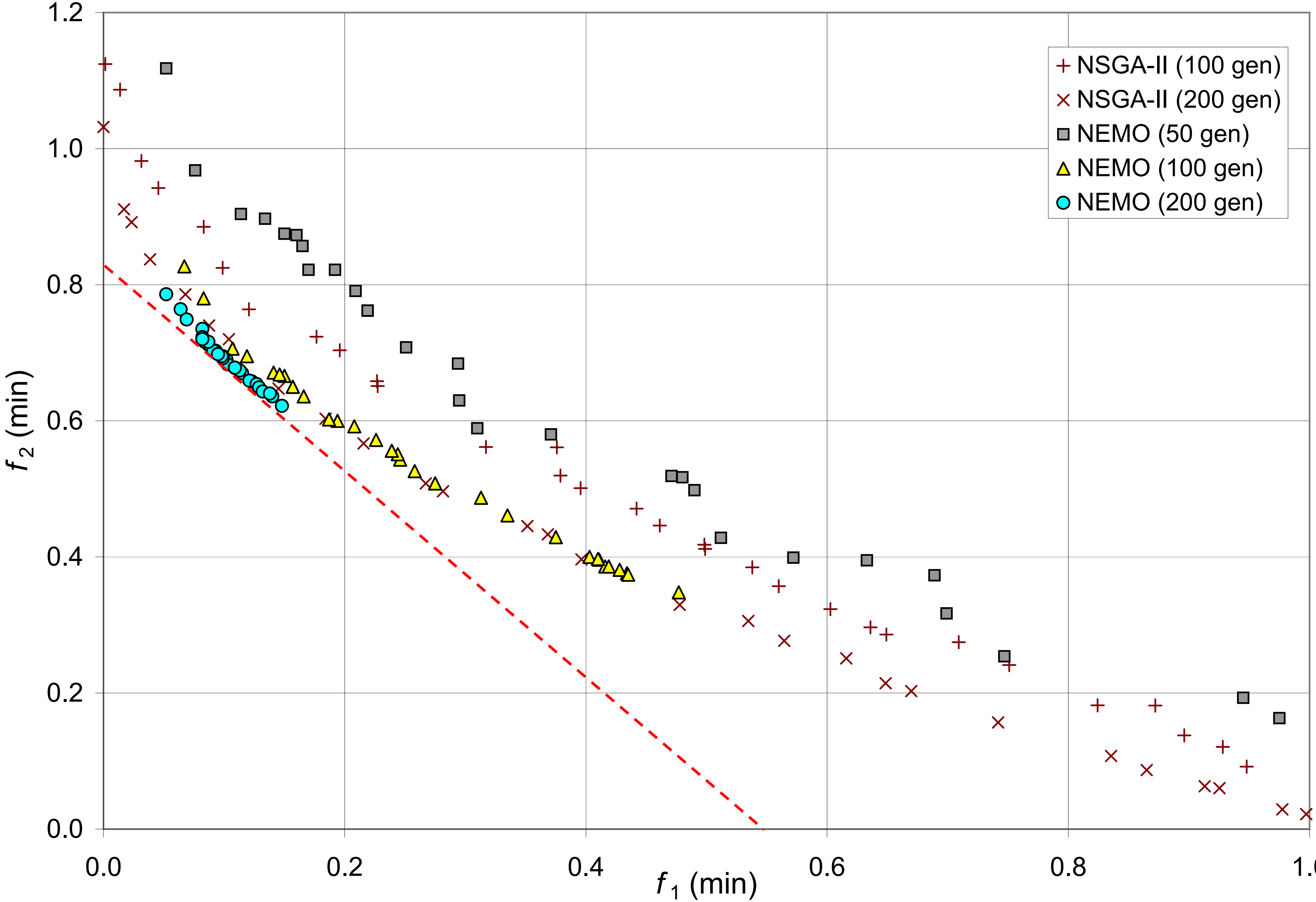


NEMO [Branke, Greco, Slowinski, Zielniewicz, 2010]

- Additive monotonic utility function

$$U(x) = \sum_{i=1}^{d} u_i(f_i(x))$$

- Pairwise comparisons to restrict set of utility functions compatible with preference information
- A is necessarily preferred over B if there is no compatible utility function that would prefer B


NSGA-II (100 gen)
NSGA-II (200 gen)
NEMO (50 gen)
NEMO (100 gen)
NEMO (200 gen)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
f_2 (min)
0.0
0.2
0.4
0.6
0.8
1.0
f_1 (min)

Uncertainty as additional objective

In many applications, uncertainty is a criterion

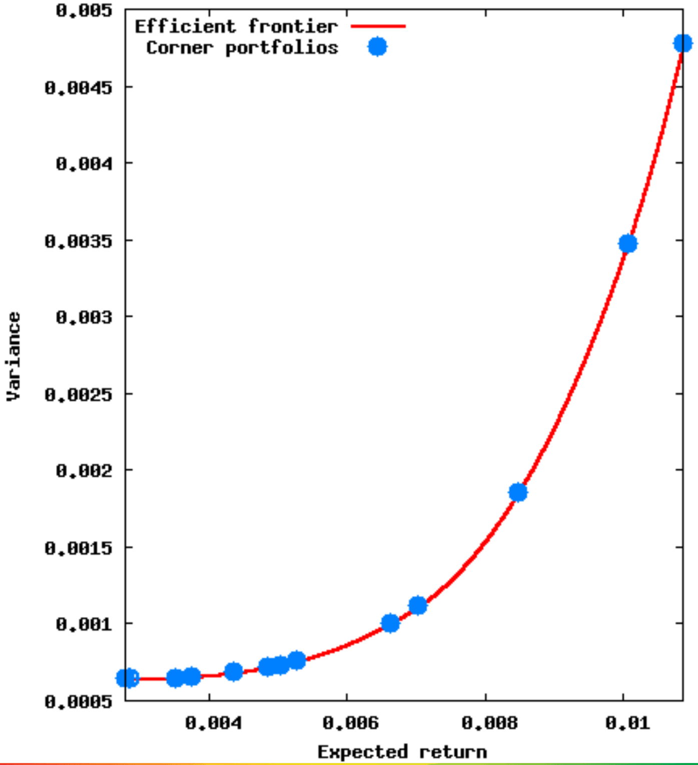
- Finance
  - maximize return
  - minimize risk (variance, VaR)

- Engineering
  - maximize performance
  - minimize probability of failure / maximize reliability

- Military
  - minimize cost
  - maximize probability of success

Portfolio optimisation

$$\begin{aligned} \text{Min} \quad V(\boldsymbol{x}) &= \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} \\ \text{Max} \quad E(\boldsymbol{x}) &= \boldsymbol{\mu}^T \boldsymbol{x} \\ \text{s.t.} \sum_{i=1}^{n} x_i &= 1 \\ \boldsymbol{A}\boldsymbol{x} &\leq \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0} \end{aligned}$$



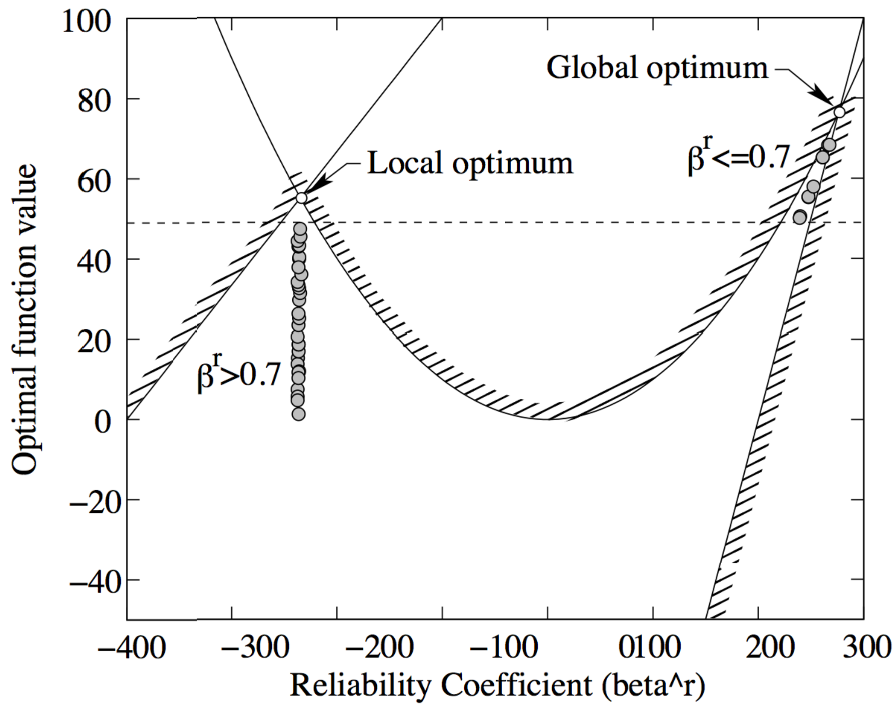
Possible challenge: how to evaluate uncertainty

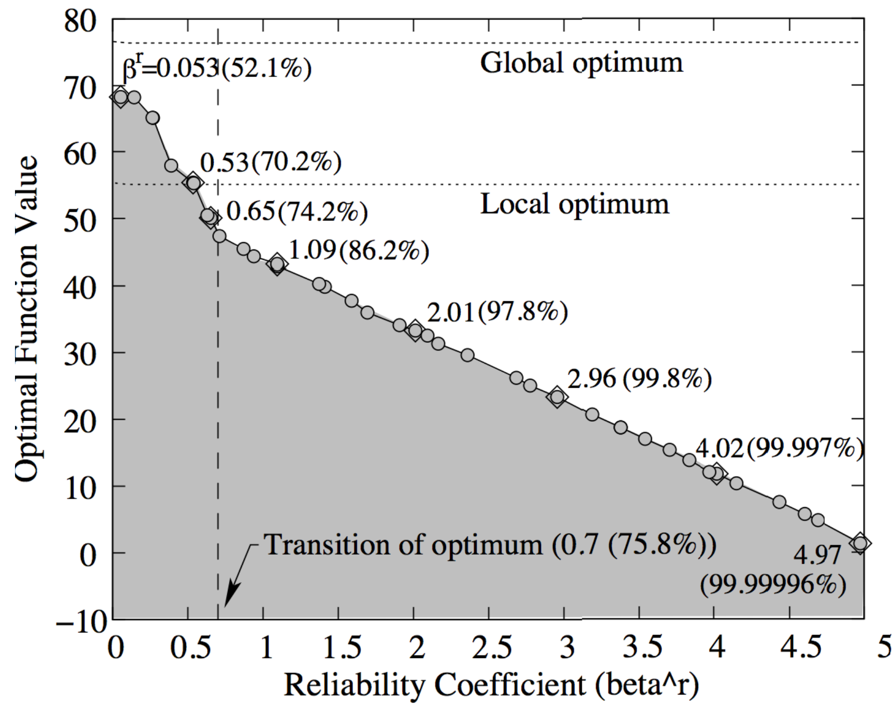
- Monte Carlo sampling
- Expected performance in local fitness space (use metamodels for estimation)
- Distance to constraint (reliability-based optimization)

Reliability-based Optimization

[Deb, Gupta, Daum, Branke, Mall, Padmanaban 2009]

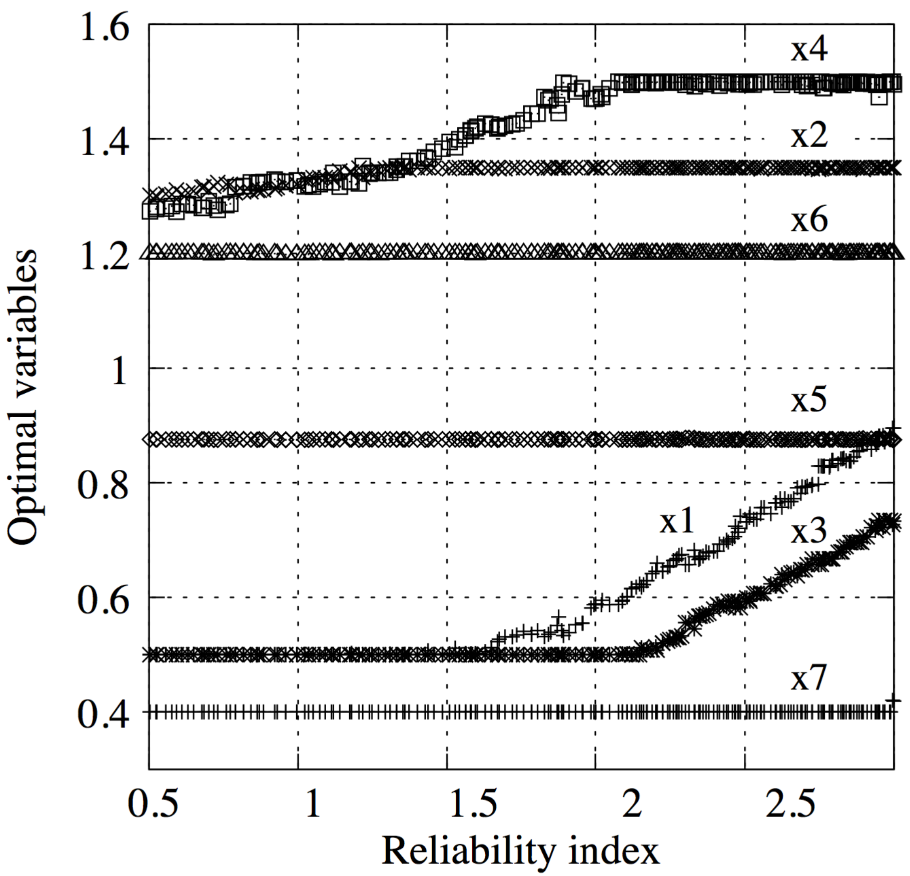
- Use distance from constraints as additional objective
- MOO allows to show possible trade-off





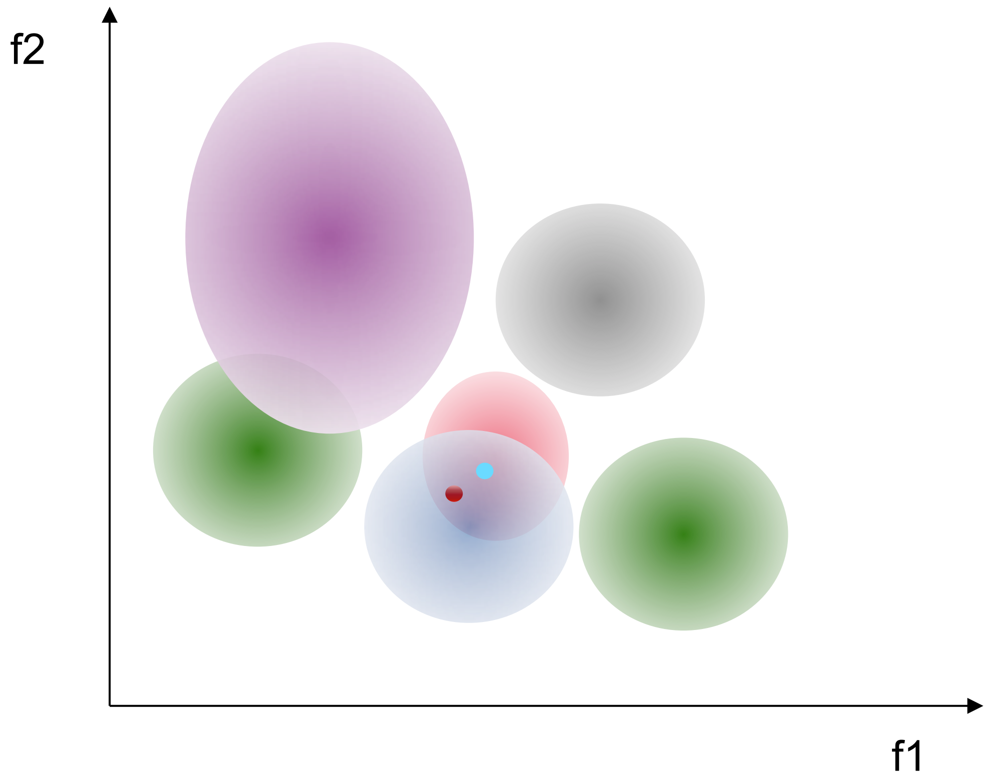
Reliability - Innovization

- Car side impact problem

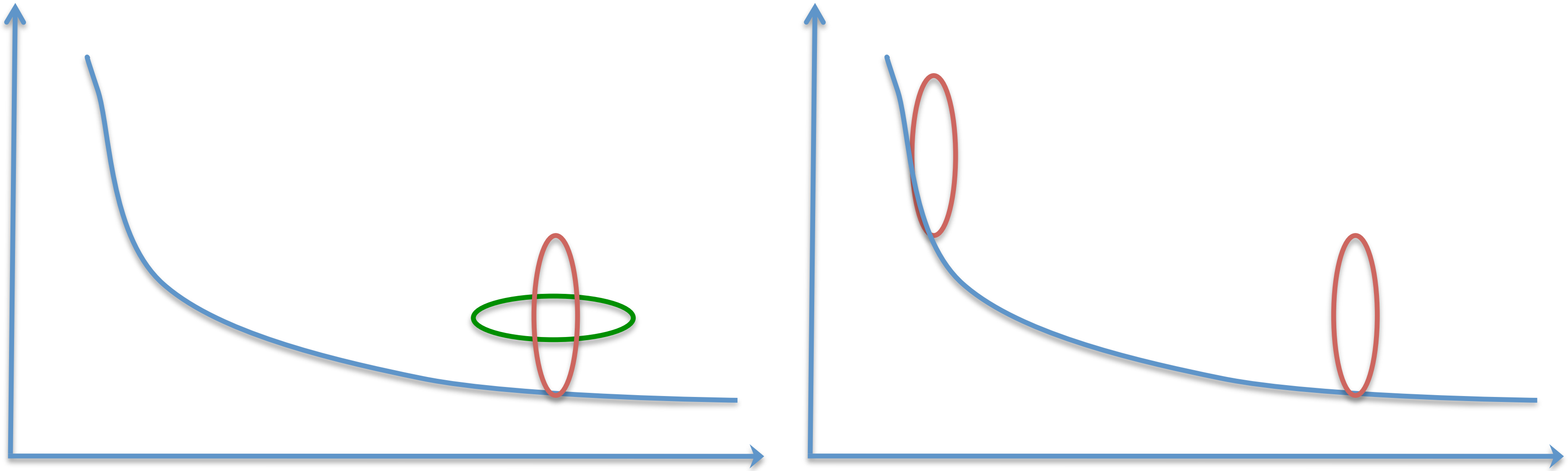


Noisy objective functions

How to compare “clouds”



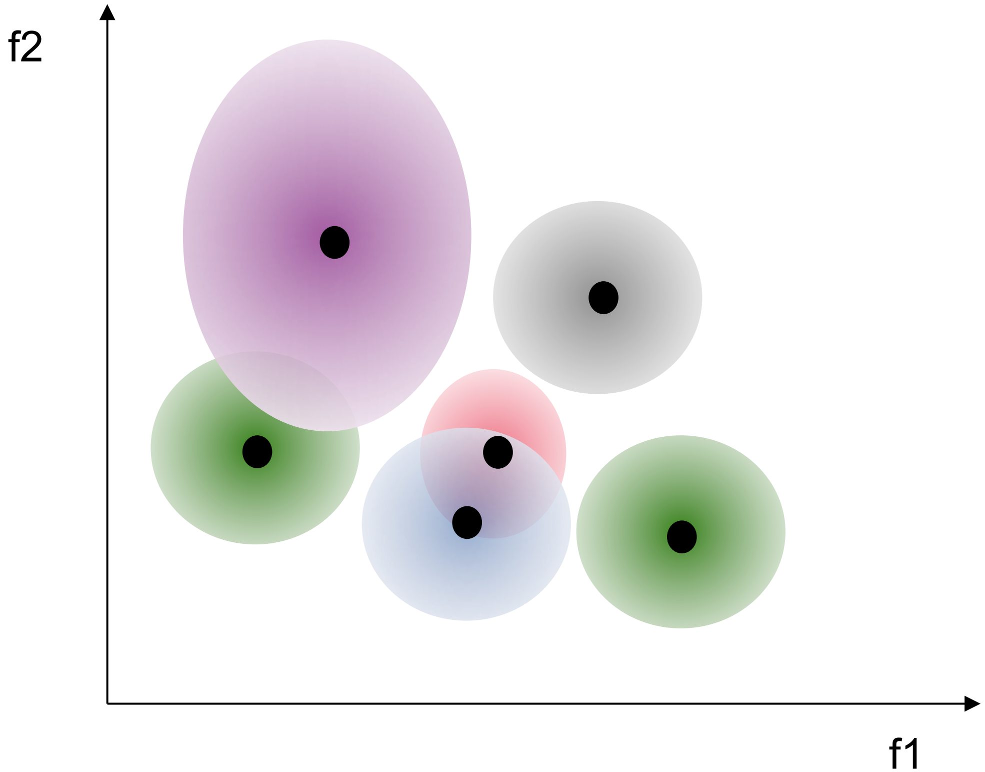
How to compare "clouds"



- Is rotation important?
- Is location important?
- Do we need to be able to scale objectives to decide on robustness?

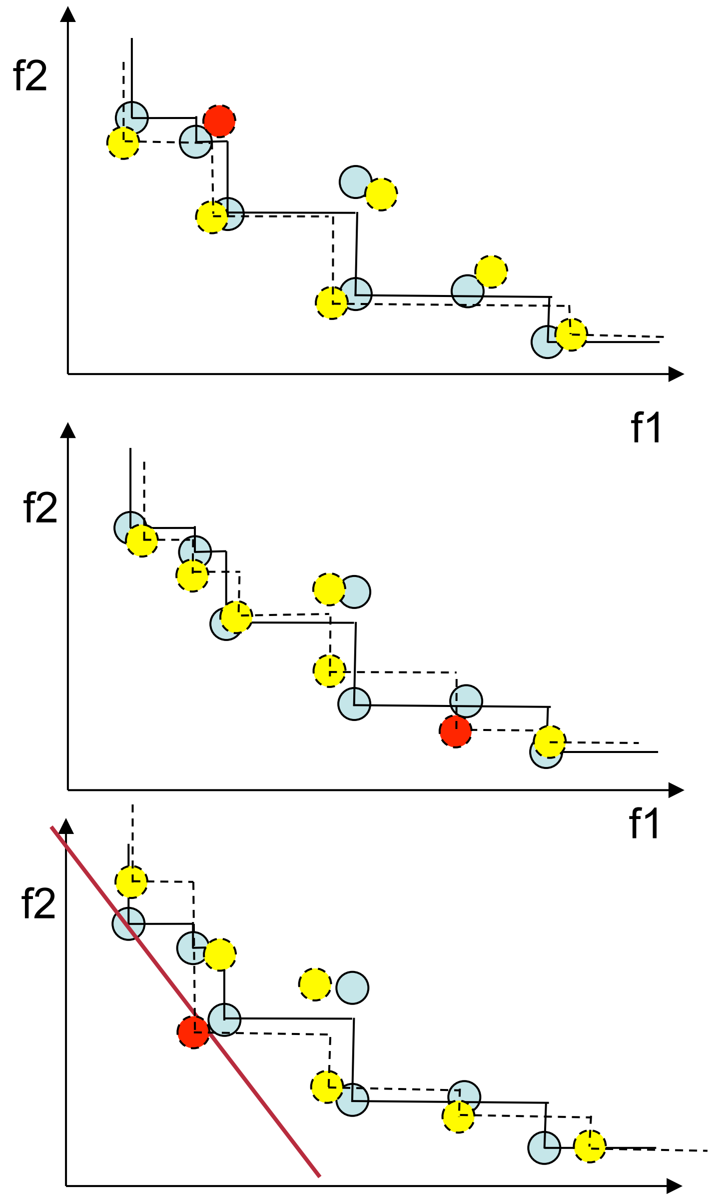
For now

Identifying the solutions with the best expected objective values

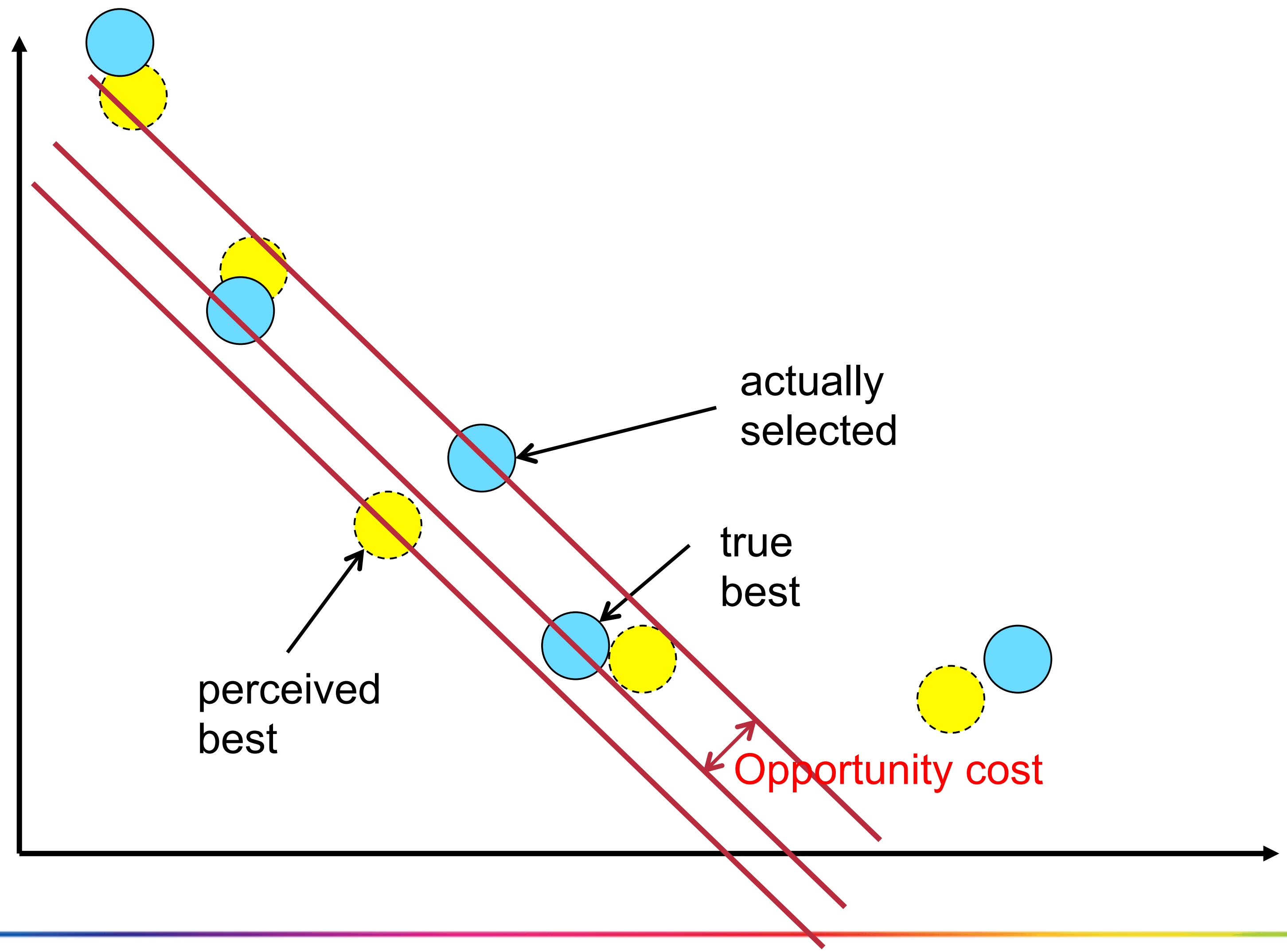


Possible errors

- A non-dominated solution is not recognized as such and thus not presented to the user.
- A dominated solution appears to be non-dominated.
- The best solution is recognized as being non-dominated, but another solution erroneously appears more attractive to the user.

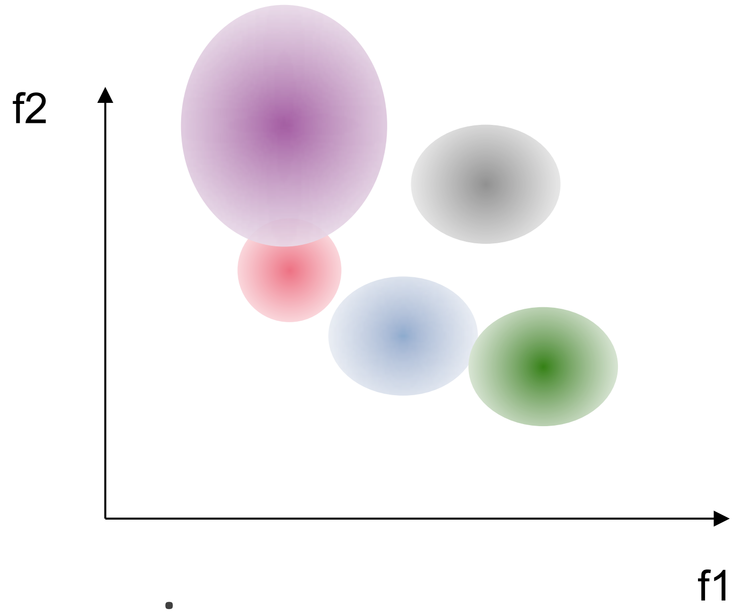


Expected opportunity cost



Sequential sampling to minimize expected opportunity cost

- Given: set of solutions, computational budget
- Goal:
  minimize opportunity cost
- The more samples, the more accurate the fitness estimate



- Samples are computationally expensive
- Start with few samples, then allocate more where needed
- Optimal Computing Budget Allocation

iMOCBA [Branke&Gamer, 2007]

- Idea: User is represented by probability distribution over λ
- Probability distribution is used to calculate overall EOC and estimate value of additional sample
- W.l.o.g., we assumed linear utility functions $U(x)=\lambda f_1(x)+(1-\lambda)f_2(x)$ and a uniform probability distribution over λ

Expected Opportunity Cost

- For a given λ and two solutions, EOC can be calculated based on probability distribution

$$EOC = \int_{x=s}^{\infty} (x - s)\phi(x)dx$$

where s is observed utility difference and φ is prob dist for difference

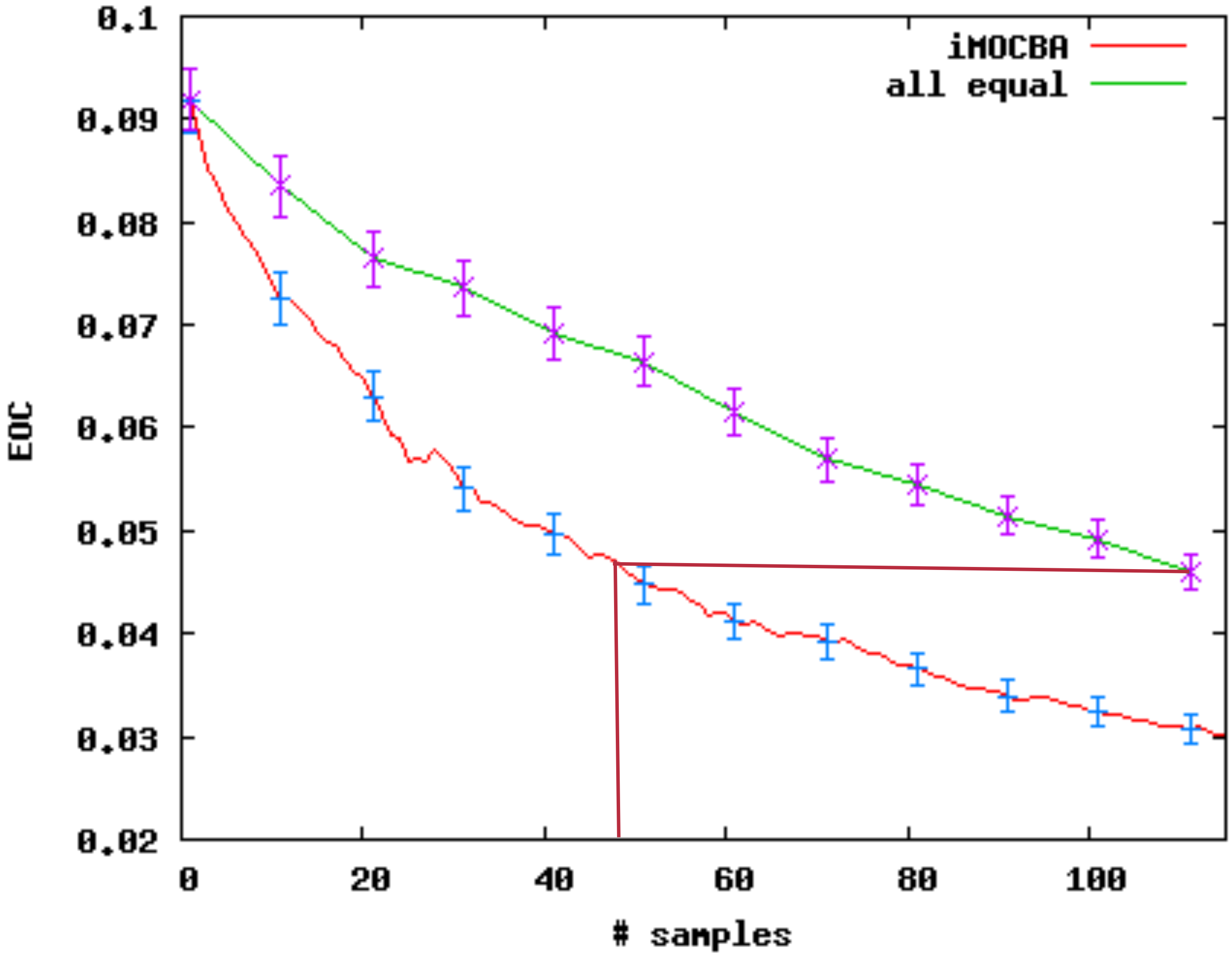
- Extension to more than 2 solutions by Bonferoni bound (sum of pairwise EOCs)
- Extension to many λ by summing up over 1000 equally-spaced λ
- Approximate effect of additional sample by effect on variance

iMOCBA

- Initialize: Sample each system n_0 times
- WHILE budget not exhausted
  - Calculate OEEOC_i, the overall estimated EOC if system i receives another sample based on probability distribution $P(\lambda)$ of user's utility function
  - Actually sample system i with i =argmin{OEEOC_j}
  - Update sample statistics
- User selects system with best perceived utility

Efficiency (1)

RPI, 10 systems



Worst-case optimisation

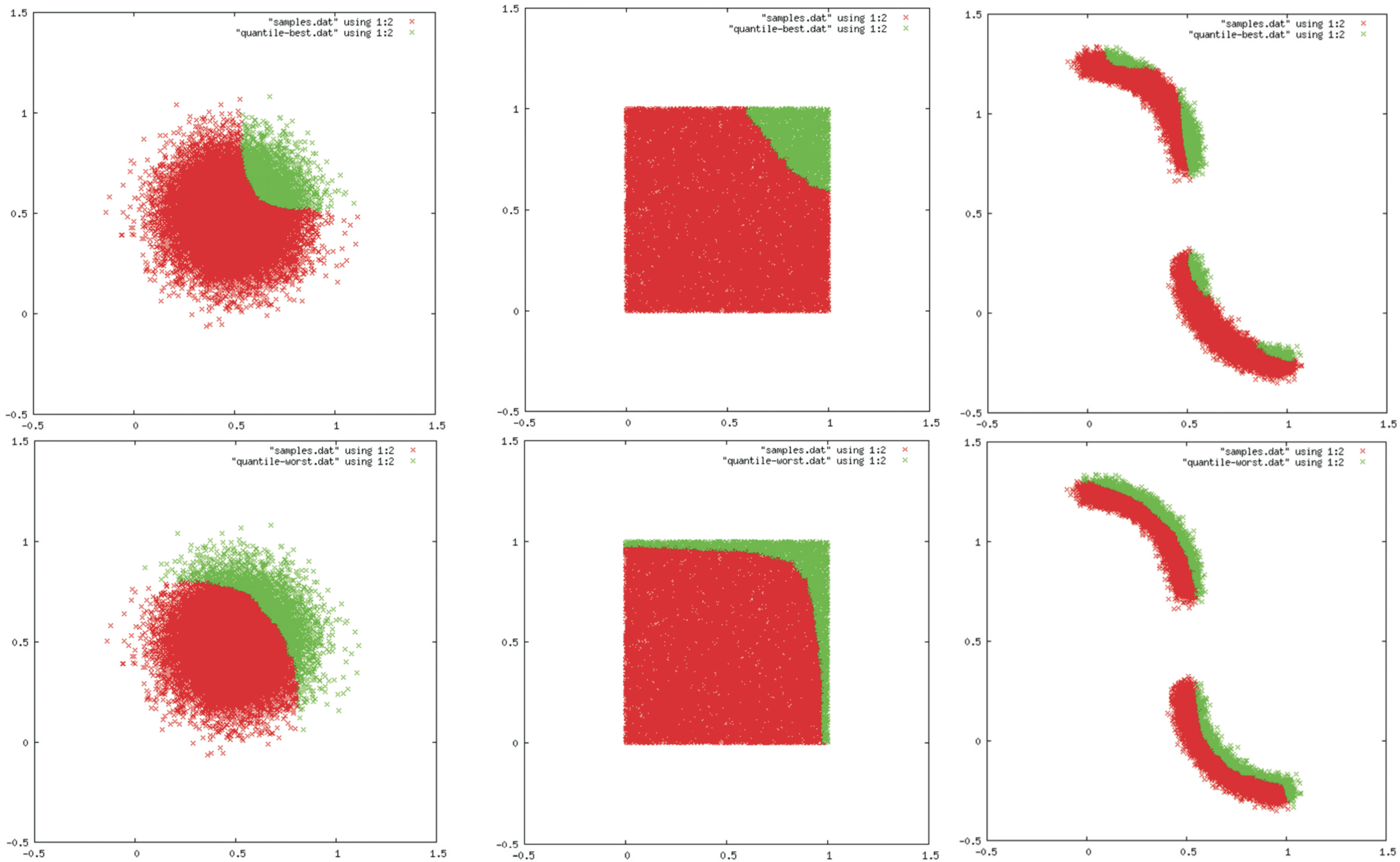
[Branke, Avigad, Moshaiov 2008]

What if DM is not risk neutral?

- Stochastic dominance P(A≥x)≥ P(B≥x) for all x
  $\Phi_A(x) \leq \Phi_B(x)$ for all x
- Quantiles
- Value at risk
- Worst case

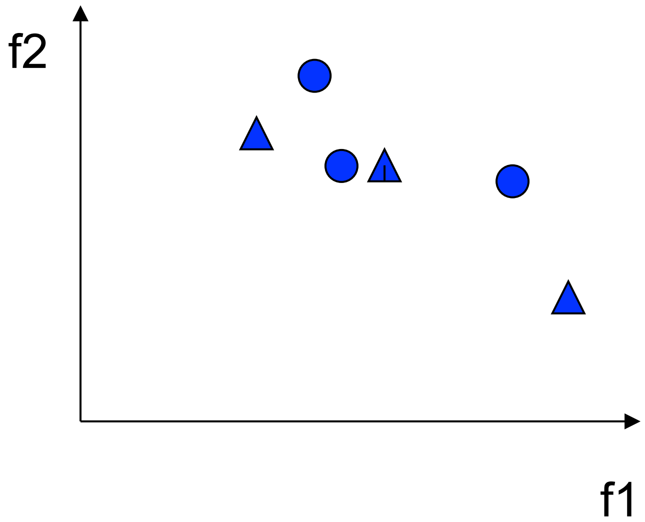
- But: all these are not defined in the case of multiple objectives!

Quantiles in MOO [Bosman 2009]



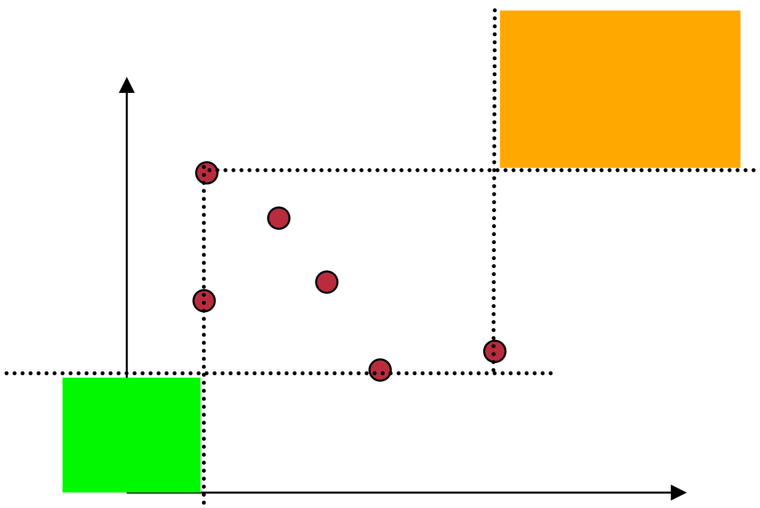
Worst case in MOO

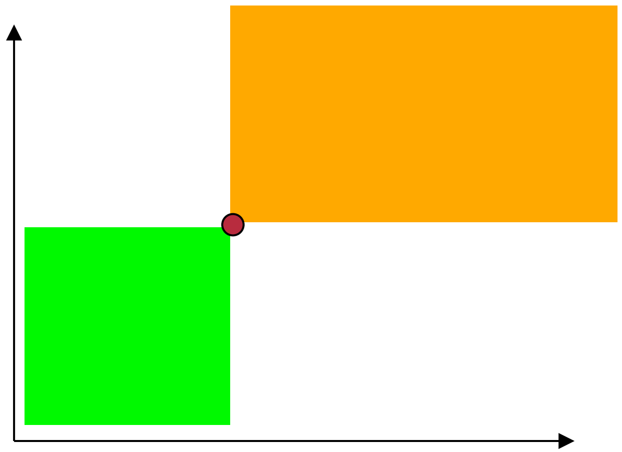
- Which scenario is worst case depends on the user's preferences



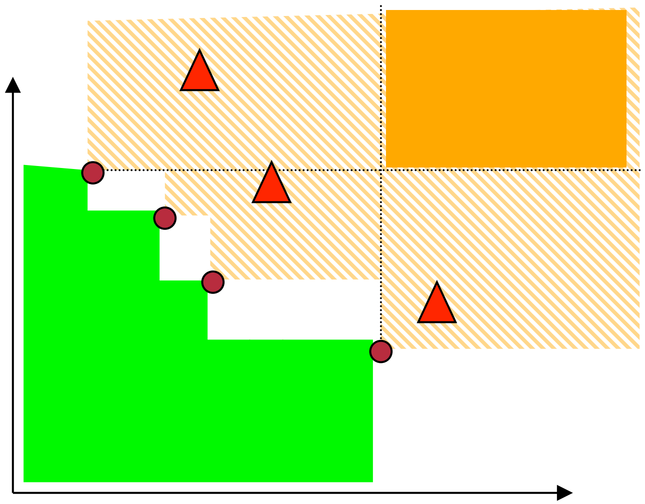
Total dominance

- Each scenario of A dominates each scenario of B



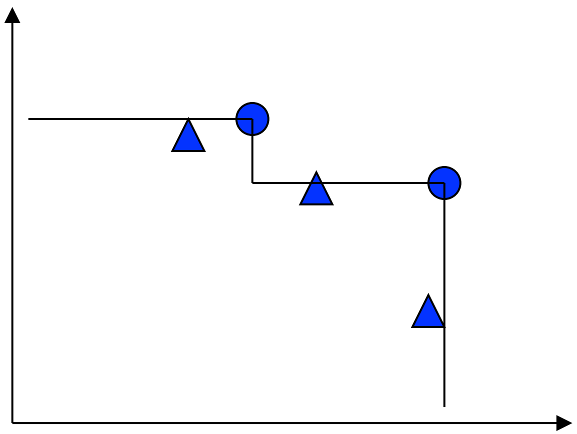


Worst-case dominance

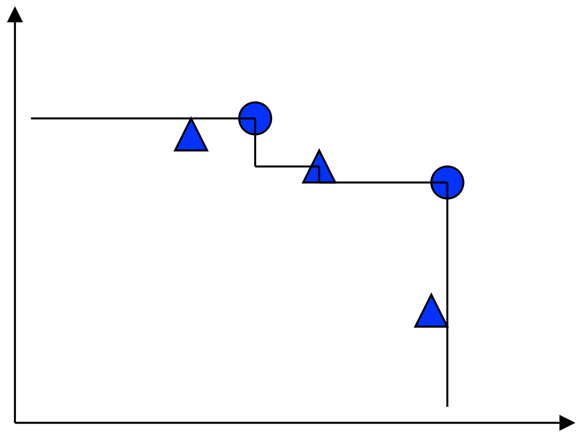


Worst-case dominance

- A solution A dominates a solution B with respect to worst case, if the non-dominated set of A ∪ B with respect to the inverted (maximization) problem only contains representatives of solution B



A dominates B



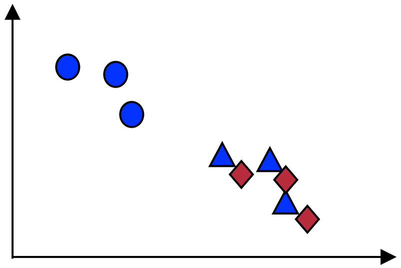
A and B are non-dominated

▲ Solution A ● Solution B

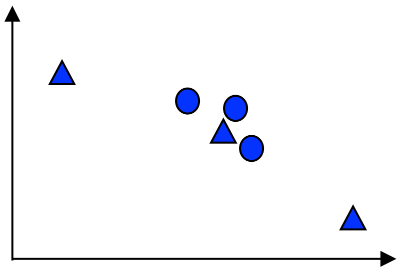
How to rank non-dominated solutions?

Three important aspects:

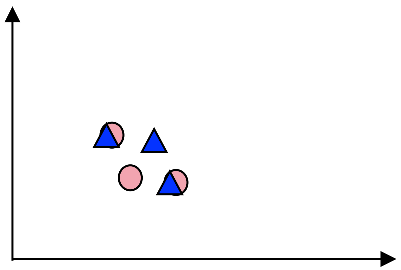
Diversity



Spread

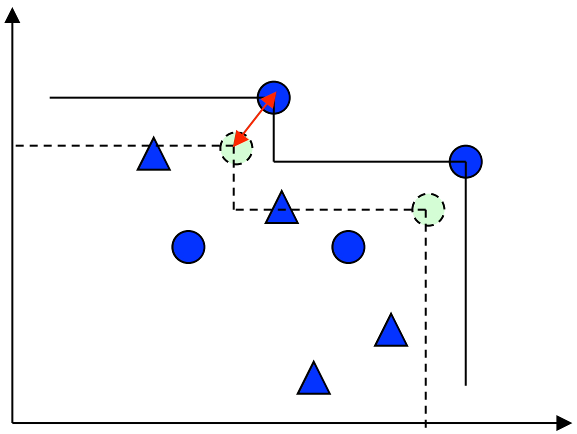


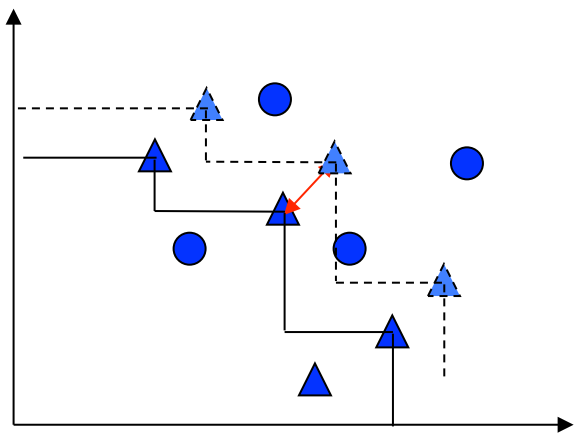
Convexity



First approach: δ-Indicator

- Solution fitness is distance a solution can be moved simultaneously in all objectives before it becomes dominated, or, if it is dominated, the minimal distance it has to be moved until it becomes non-dominated



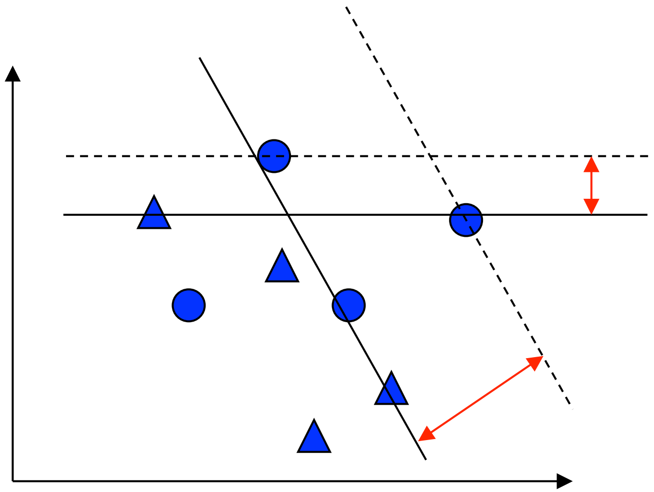


Second approach: Marginal expected utility

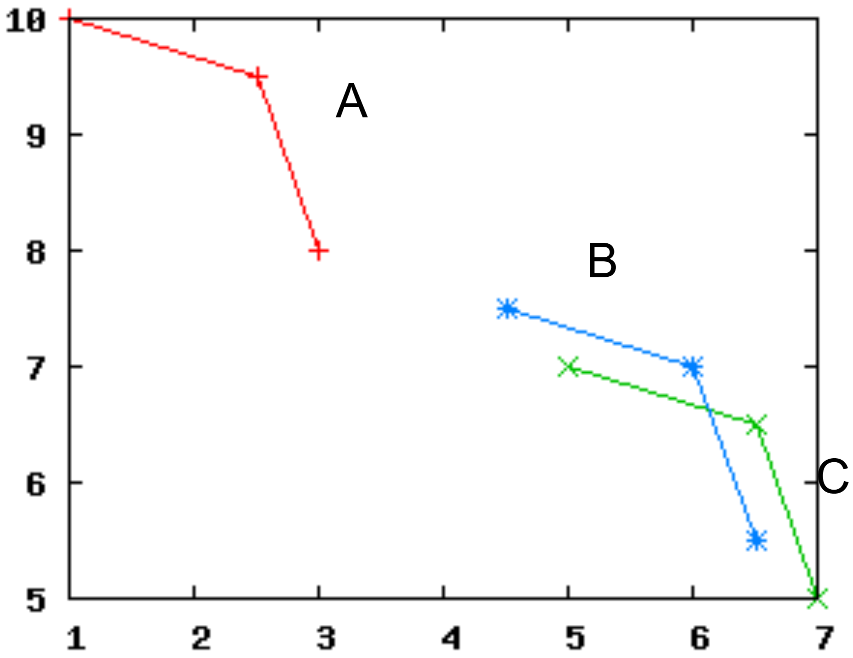
- Assume linear utility function $u=-(\lambda f_1+(1-\lambda)f_2)$
- Calculate marginal utility by numerical integration over λ, assuming uniform distribution in [0:1]
- For each λ and solution *i*, let $w(\lambda,i)$ be the worst case utility
- To calculate marginal utility u':
  - Set marginal utility $u'(i)$ of all solutions to zero
  - For each λ find best and second best solution i^* and i'
  - $u'(i^*) \leftarrow u'(i^*)+w(\lambda,i^*)-w(\lambda,i')$

Marginal expected utility (2)

Illustration:

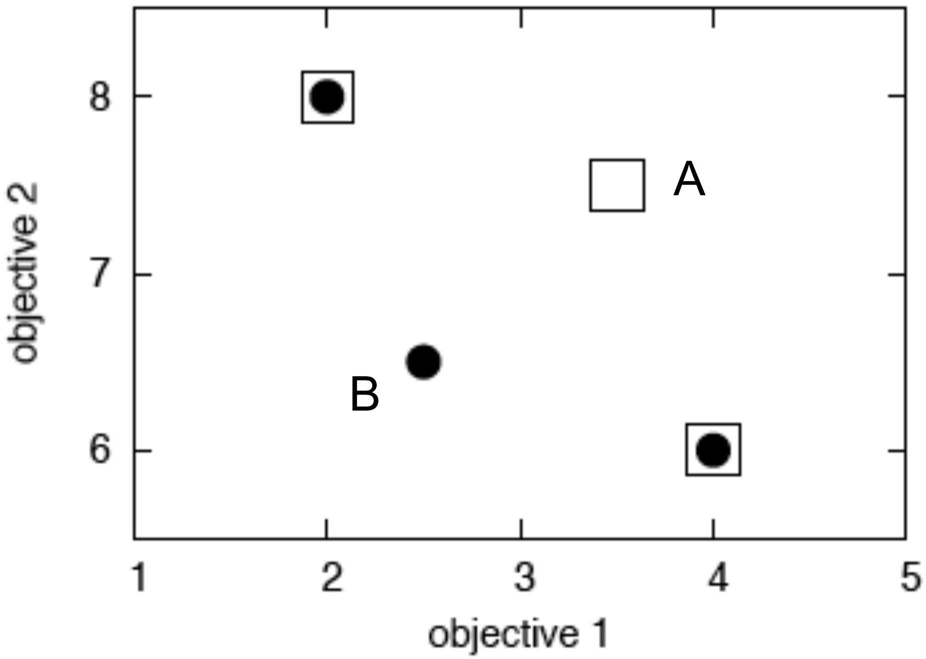


Example Diversity



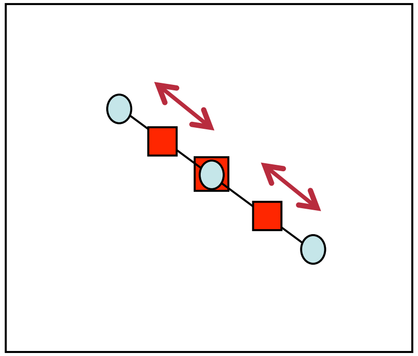
- δ indicator:
  - A 3.5
  - B 0.5
  - C 0.5
- Marginal utility:
  - A 100
  - B 0
  - C 12

Example Convexity

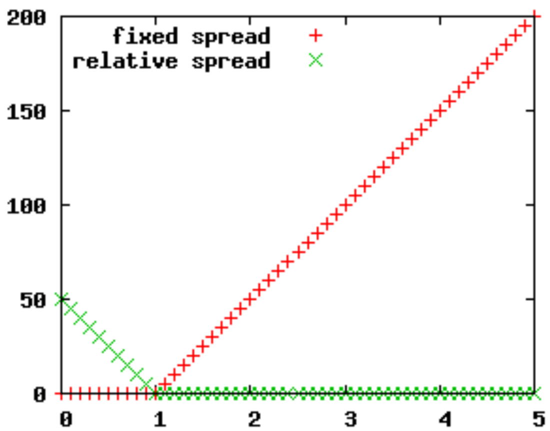


- δ indicator:
  - A 0
  - B 1
- Marginal utility:
  - A 0
  - B 1

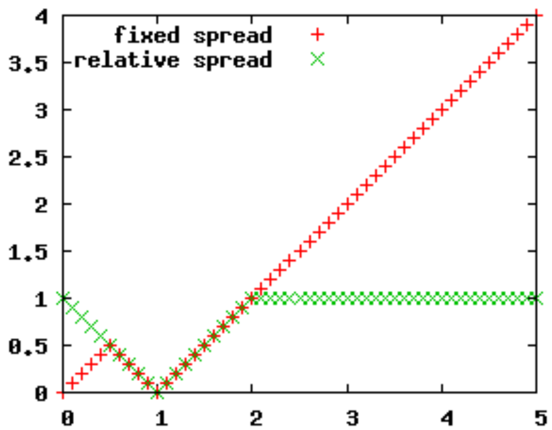
Example Spread



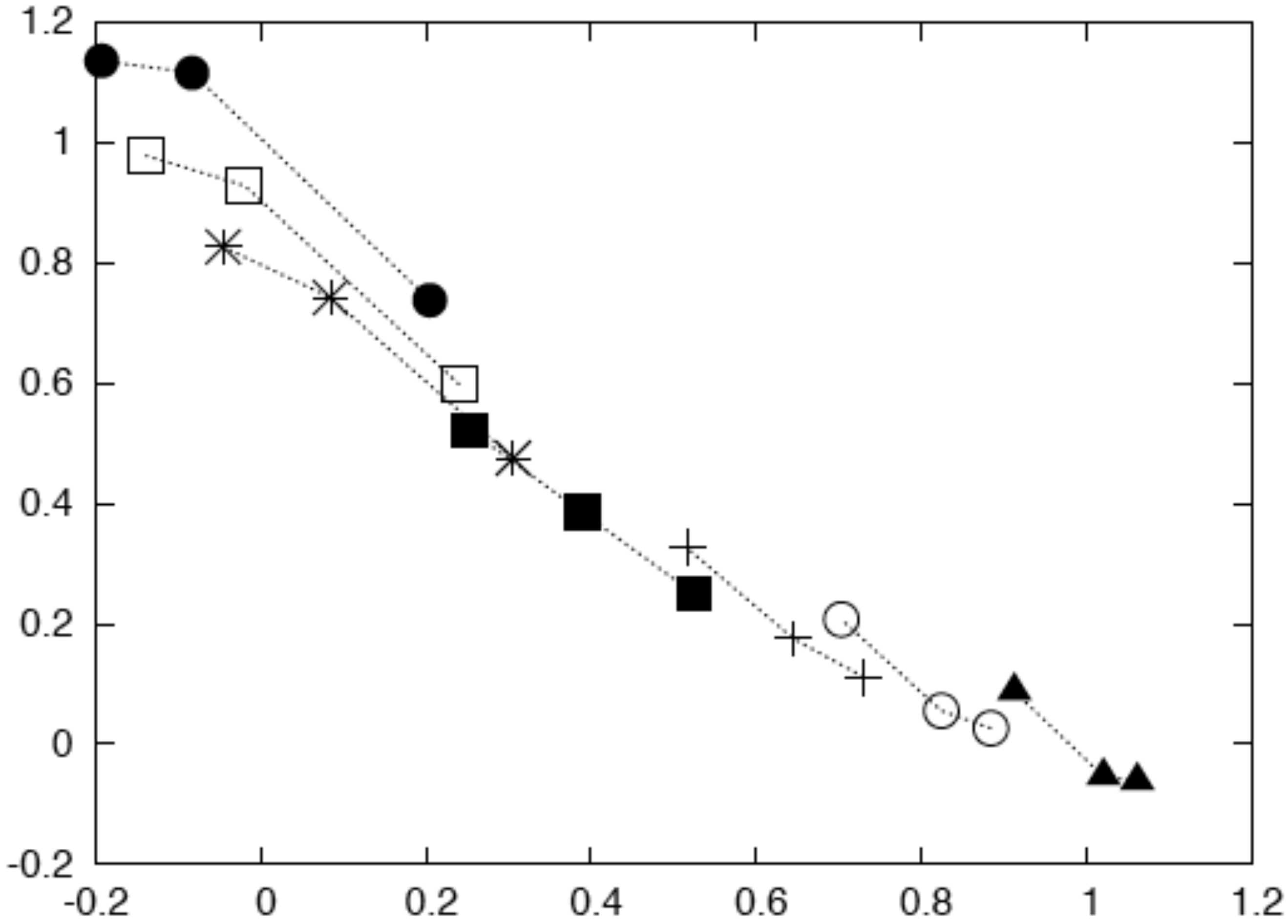
Marginal utility



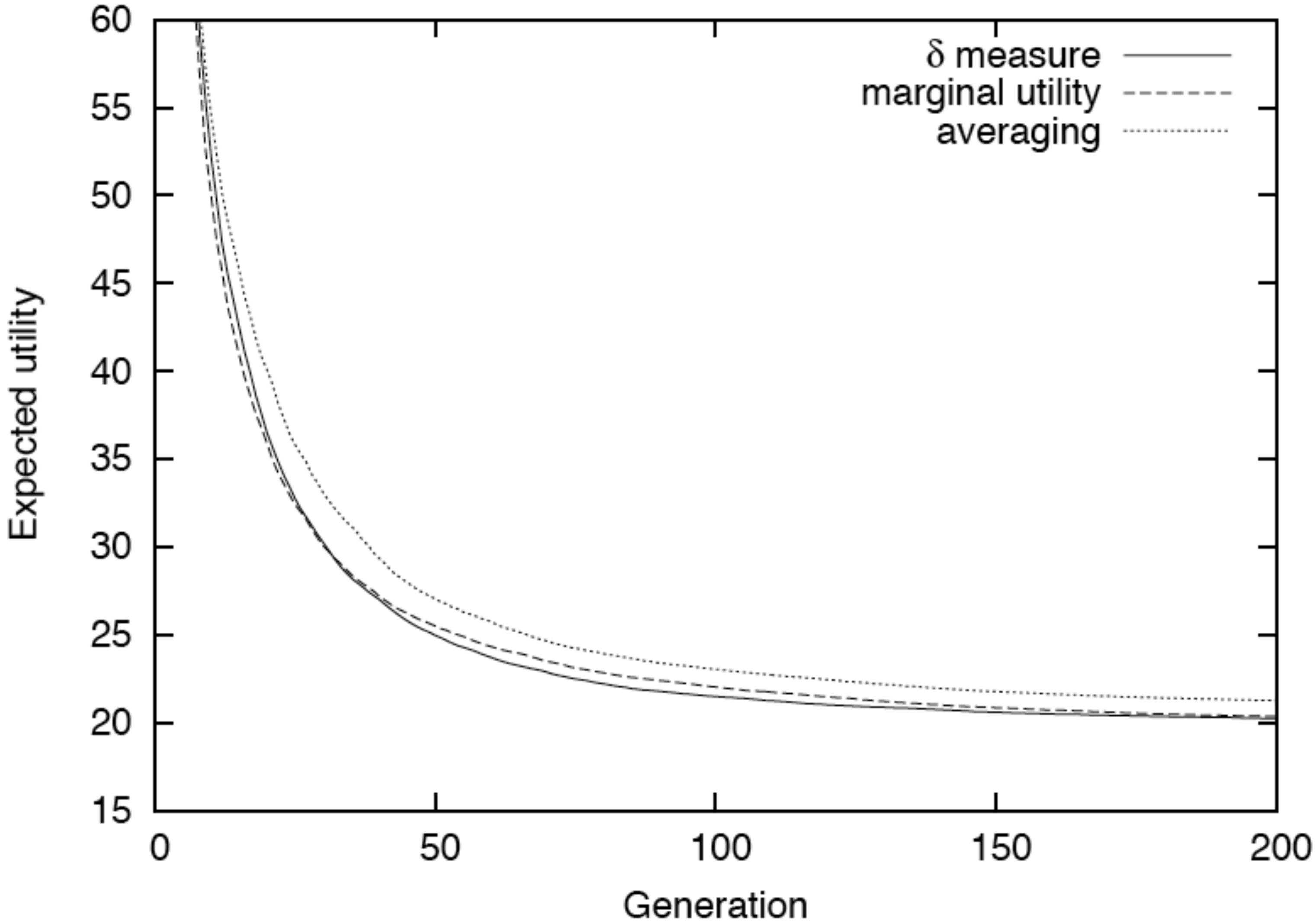
delta-measure



Artificial test problem



Expected utility



Summary

- Different aspects of uncertainty
  - user preferences
  - uncertainty as additional objective
  - uncertainty in objective function values
- Many concepts from single-objective optimisation don't translate to multiple objectives – what do we want to achieve?
- Notion of probability distribution over utility functions seems quite useful

Special session

Multiobjective optimization and decision making under uncertainty

17-21 June 2013

Any uncertainties left?