Meta-Model Assisted (Evolutionary) Optimization

Tutorial at PPSN 2016 - 18.09.2016

Boris Naujoks, Jörg Stork, Martin Zaefferer, Thomas Bartz-Beielstein

Technology
Arts Sciences
TH Köln

SYNERGY

Institut "Jožef Stefan"
Ljubljana, Slovenija

Université
de Lille
SCIENCES
ET TECHNOLOGIES

Technology
Arts Sciences
TH Köln

Intro and Motivation

- Synonyms
    - Metamodels
    - Surrogates
    - Response surface models
    - Approximation models
    - Simulation models
    - Data-driven models
    - Emulators

- From Latin surrogatus
    – a replacement for something, a substitute or alternative
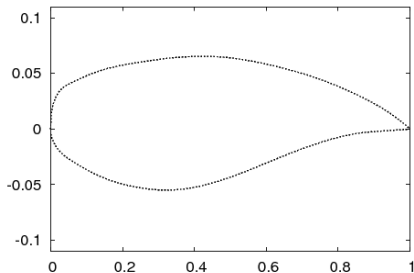
    Perfect passive participle of surrogare
    - Variant of subrogare, from
    - Sub (under) + rogare (ask)

Most common applications

- Engineering design
- Long, expensive fitness function evaluations
  - Finite elements models
  - Computational fluid dynamics models
- Examples
  - Airfoil design
  - Ship propulsion systems
  - etc.

Most common applications

- Engineering design
- Long, expensive fitness function evaluations
  - Finite elements models
  - Computational fluid dynamics models
- Examples
  - Airfoil design
  - Ship propulsion systems
  - etc.

Most common applications

- Engineering design
- Long, expensive fitness function evaluations
  - Finite elements models
  - Computational fluid dynamics models
- Examples
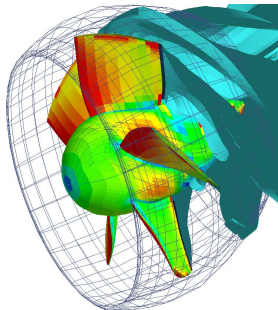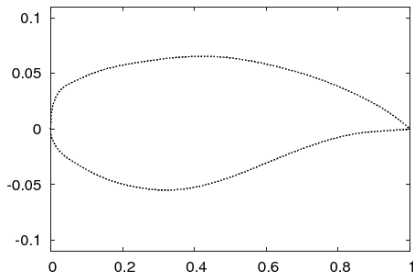 - Airfoil design
  - Ship propulsion systems
  - etc.

Image taken
from [Naujoks et al., 2007]

Other areas

. . . where surrogates are applied, involved, used . . .

- No explicit fitness function available
  - Fitness depending on external factors, e.g. human interactions
  - Music and arts

- Uncertain environments
  - Noisy environments
  - Robustness wrt. design variables
  - Dynamic fitness landscapes

- Smoothing multi-modal fitness landscapes

Overview

- Motivation

- Concepts and methods

- Practical approach: instructive application

- Typical problems in application

- Open Issues / Research perspectives / Fields of Interest
  - Multi-criteria optimization
  - Combinatorial optimization

- Discussion
  - Typical problems and their solutions

Overview

- Motivation

- **Concepts and methods**

- Practical approach: instructive application

- Typical problems in application

- Open Issues / Research perspectives / Fields of Interest
  - Multi-criteria optimization
  - Combinatorial optimization

- Discussion
  - Typical problems and their solutions

Surrogate Modeling - Concepts and Methods

Questions to Answer:

1. What is the core concept of surrogate modeling?
2. How does a typical surrogate optimization cycle work?
3. Which models are common for surrogate optimization?
4. Example method: *Efficient Global Optimization*

Costly real world (blackbox) problems

- Real-world applications: commonly blackbox problems (machines, complex processes)
- Available information is very sparse, properties of the objective function are difficult or impossible to determine,
- No *a priori* information about modality, convexity, gradients, or the minimal function value $f(x^*)$ is known
- Most complex problems arise if physical experiments are involved. Aside from being costly in terms of needed resources (manpower, material, time)
- Wrong values can lead to hazardous effects, e.g., damaging or destroying experimental material.
- Instead of physical experiments, simulations are used, e.g., from the field of Computational Fluid Dynamics (CFD).
- Require a lot of computational power and are very time demanding

Inevitable need to evaluate candidate solutions in the search space to retrieve any information and high demand on resources for each of these evaluations.

Surrogate Modeling - Application Layers

L1 The Real-World Application
- Direct optimization is very costly or impossible as incorrectly chosen decision variable values
- Evaluations involve resource demanding prototype building or even hazardous experiments

L2 The Simulation Model
- Complex computational model from fluid- or structural dynamics
- Single simulation process may take minutes, hours, or even weeks to compute
- Available computational power limits the amount of available evaluations
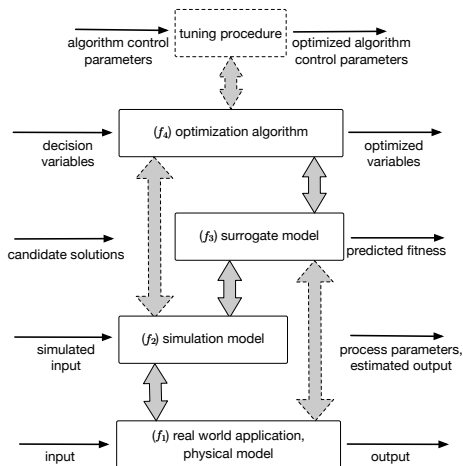
L3 The Surrogate Model
- Data-driven regression model
- The accuracy heavily depends on the underlying surrogate type and number of available information
- Typically cheap

L4 The Optimization Process
- Any suitable optimization algorithm (deterministic, stochastic, metaheuristic...)
- Can be tuned

Surrogate Modeling - Core Concept

Surrogate Modeling - Costs and Benefits

- Each layer L1 to L4 imposes different evaluation costs and solution accuracies:
- Most expensive: L1 real world
- Commonly cheapest: L2 Surrogate Model
- Modeling process itself requires computational resources for evaluations, construction or validation of the surrogate.

The main benefit of using surrogates is the reduction of needed fitness evaluations on the objective function during the optimization.

- An other advantage is the availability of a surrogate itself, which can be utilized to gain further problem insight.
  -> This is particularly valuable for blackbox problems.
- The initial sampling design plan has a major impact on the optimization performance and should be carefully selected.

Surrogate Modeling - Optimization Cycle

A common optimization process using surrogates is outlined by the following steps:

1. Sampling the objective function to generate a set of evaluated points
2. Selecting a suitable surrogate
3. Constructing the surrogate using the evaluated points
4. Utilizing the surrogate to predict new promising locations
5. Evaluating the objective function on one (or more) of the identified locations
6. Updating the surrogate and repeating the optimization cycle

Surrogate Modeling - Important Publications

Important publications featuring overviews or surveys on surrogate modeling and surrogate optimization:

- Design and analysis of computer experiments, [Sacks et al., 1989]
- A taxonomy of global optimization methods based on response surfaces, [Jones, 2001]
- Surrogate-based analysis and optimization, [Queipo et al., 2005]
- Recent advances in surrogate-based optimization, [Forrester and Keane, 2009]
- Surrogate-assisted evolutionary computation: Recent advances and future challenges, [Jin, 2011]

Linear Models

- Combination of linear predictor functions of each input to model the output
- Basic LM: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$, where ε is the error term
- Extensions: Interactions between inputs, quadratic terms, response surface models, polynomial regression
- Polynomial model takes the form
  $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n + \varepsilon$

Pro: white box, easy to interpret / analyse, simple and fast

Con: not suitable for complex functions, overfitting (by using too many terms)

Decision Trees and Random Forests

- Decision Trees [Breiman et al., 1984] model the objective function by using tree-based approximations.
- At each node of the tree a split is made on the basis of an decision variable value
- The prediction of a new point is given by the mean value of associated points
- Random Forests Regression [Breiman, 2001] a large number of decision trees is combined to an ensemble predictor
- Usually, each tree in the ensemble is fitted using a subset of the evaluated points to avoid overfitting (*bagging*)
- Predictions of new individuals are then given by a cumulated mean of all predictors in the ensemble

Pro: easy to interpret white box model (decision trees), fast, binary+integer+real variables

Con: complex to interpret (RF), bad fit for complex functions (decision tree), no smooth surface, overfitting (too large tree)

Artificial Neural Networks and Deep Learning

- Neural Networks [Haykin, 2004; Hornik et al., 1989] are inspired by the biological brain
- They utilize so-called connected neurons to learn and approximate the behavior of a function
- Neurons are weighted transform functions
- Several layers of neurons: input, output and hidden layers
- Layers consist of neurons with different forward an/or backward connections
- Deep learning [Deng and Yu, 2014; Hinton et al., 2006]:
- Complex structured networks with multiple processing layers and/or multiple non-linear transformations and stacked model approaches
- Excellent results in approximation and specially classification tasks
- Highly computational complex, lot of resources needed

Pro: very accurate (deep learning), universal approximator

Con: high computational effort, difficult to interpret, very complex (Deep Learning)

Symbolic Regression

- Symbolic Regression [Flasch et al., 2010] is a high level method to fit a human-readable mathematical model
- Based on Genetic Programming (GP)
- Mathematical expressions building blocks $(+, -, \sin, \cos, \exp ...)$
- Model is evolved using an evolutionary population-based approach

Pro: easy to interpret, fast prediction

Con: high computational complexity (building process)

Kriging

- Kriging or Gaussian Process Regression [Sacks et al., 1989] is used to model the error term of the model instead of the linear coefficients
- Simplest form: $\beta_0 + \varepsilon$, where β_0 is the mean
- The ε is then expressed by an gaussian stochastic process.
- Modeling of the error term ε with help of a covariance distance matrix
- The correlation between errors is related to the distance between the corresponding points
- The covariance matrix is utilized to predict unknown candidates.
- Outstanding feature of Kriging models: uncertainty measure for the prediction and Expected Improvement (EI):
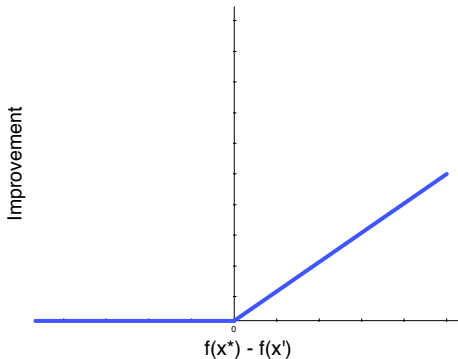
Pro: suitable for complex functions, uncertainty measurment and EI

Con: not suitable for high dimensional data, high computational effort

Expected Improvement

- Current best point for minimization: x^* with function value $f(x^*)$
- For a new point x', the improvement in our objective function is $[f(x^*) - f(x')]^+$
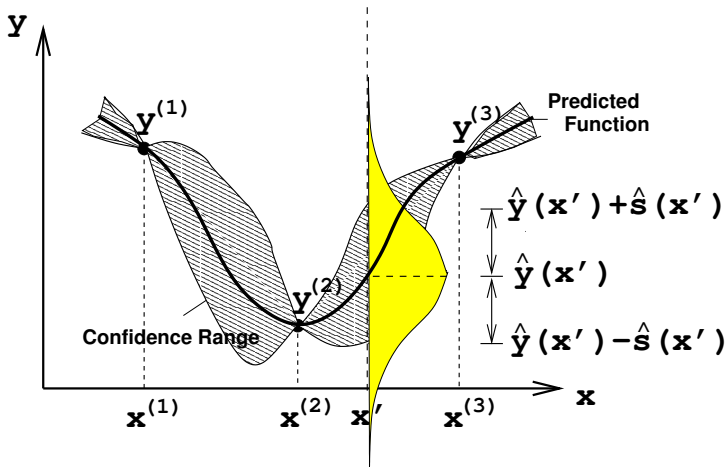
Expected Improvement

Image taken from [Emmerich et al., 2006]

Intuition: Expected Improvement is every possible improvement value weighted by its probability

Method Example: Efficient Global Optimization

- Efficient Global Optimization (EGO) by [Jones et al., 1998a] is a surrogate optimization framework specialized on utilizing Kriging and expected improvement

- Focus on optimization of expensive blackbox functions

- The original version of EGO starts by sampling the objective function by a space-filling experimental design

- Example: LHD with approximate $k = 10n$ points: convenient, finite-decimal value for the inter-point spacing, e.g., 21 design points for 2-dimensions

- Kriging surrogate is fit using maximum likelihood estimation on the selected points

- The surrogate is then manually analyzed by applying different diagnostic tests

- If it is satisfactory the iterative optimization process is started, if not, the objective function is tried to be transformed (by log or inverse transformation) to acquire a better fit.

Method Example: Efficient Global Optimization

The optimization cycle has following steps:

1. Calculate and maximize expected improvement on surrogate by a exact branch-and-bound algorithm
2. Sample the objective function where expected improvement is maximized
3. Re-estimate the Kriging surrogate including the new candidate by maximum likelihood estimation

The authors introduce a stopping criterion, which is reached if the expected improvement is less than one percent of the current best candidate

EGO Pseudo-Code Phase I: Building

Algorithm 1.1: EGO

---

begin

    phase 1, initial surrogate building:

    initialize population X of size k based on a space-filling DOE

    evaluate X on $f(x)$

    $x_c =$ best candidate in f(X)

    fit Kriging surrogate model f_m with X by maximum likelihood estimation

    manually verify f_m by diagnostic tests

    if *verify(f_m)=false* **then**

        | transform $f(x)$ by log or inverse and repeat fitting process

    end

end

EGO Pseudo-Code Phase II: Optimization

Algorithm 1.2: EGO

---

begin
    phase 2, use and refine surrogate:
    while *not* termination-condition do
        x_{new} = calculate and maximize EI on surrogate model by branch-and-bound optimization
        if $EI(x_{new})/|f(x_c)| < 0.01$ **then**
          | stop algorithm
        end
        evaluate $f(x_{new})$
        add x_{new} to X
        x_c = best candidate in $f(X)$
        re-estimate f_m with X by maximum likelihood estimation
    end
end

Overview

- Motivation

- Concepts and methods

- **Practical approach: instructive application**

- Typical problems in application

- Open Issues / Research perspectives / Fields of Interest
    - Multi-criteria optimization
    - Combinatorial optimization

- Discussion
   - Typical problems and their solutions

A practical example

- language: R
- installation of R and more: `https://cran.r-project.org/`
- optional / recommended IDE: RStudio `https://www.rstudio.com/`
- R tutorial:
  `https://cran.r-project.org/doc/manuals/r-release/R-intro.html`
- Uses a 1-dim benchmark function from Forrester et al. [2008]
- See following code

```r
## To install the required packages, uncomment the following lines:
# install.packages("SPOT")
library("SPOT") #load required package: SPOT

## Initialize random number generator seed. Reproducibility.
set.seed(1)

## Define objective function
objectFun <- function(x){
  (6*x-2)^2 * sin(12*x-4)
}

## Plot the function:
curve(objectFun(x),0,1)
```
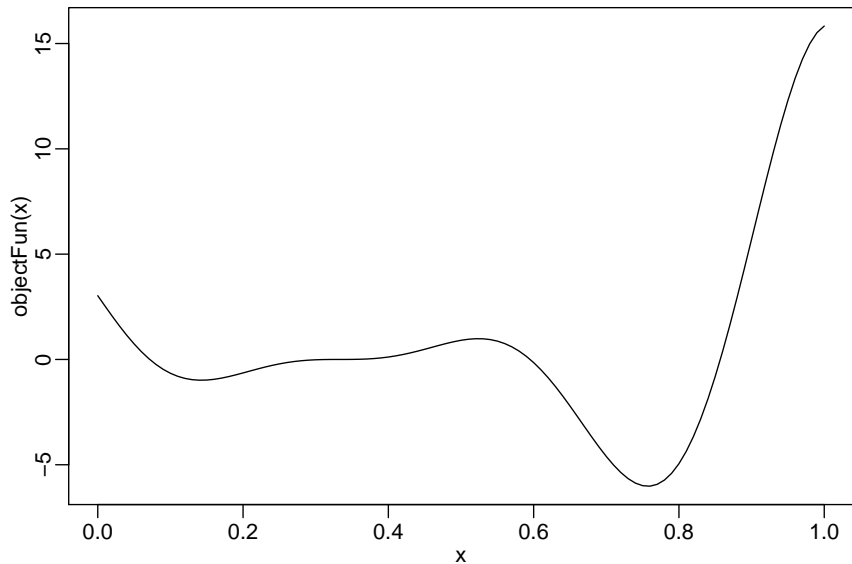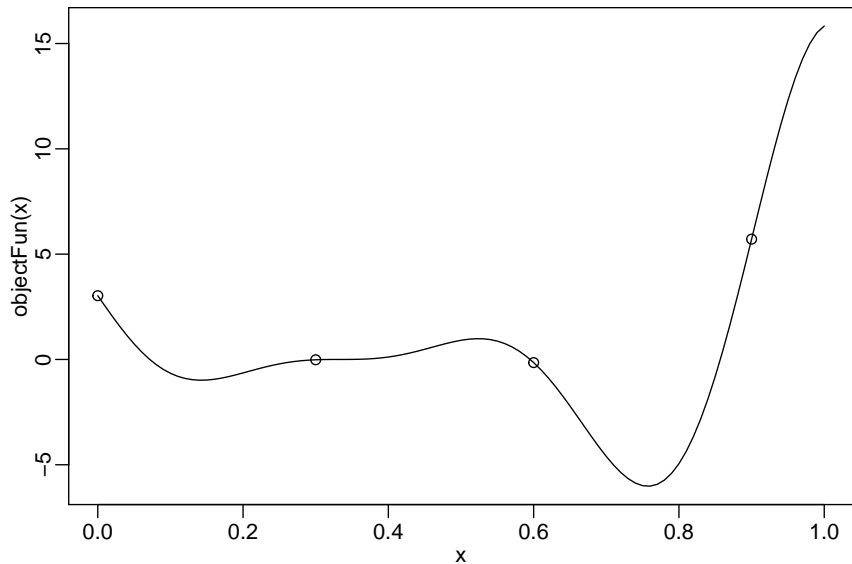
```r
## Now, let us assume objectFun is expensive.
## First, we start with making some initial
## design of experiment, which in this case
## is simply a regular grid:
x <- seq(from=0, by=0.3,to=1)

## Evaluate with objective:
y <- sapply(x,objectFun)

## Add to plot:
curve(objectFun(x),0,1)
points(x,y)
```

```r
## Build a model (here: Kriging, with the SPOT package.
## But plenty of alternatives available)
fit <- forrBuilder(as.matrix(x),as.matrix(y),
    control=list(uselambda=FALSE #do not use nugget effect (regularization)
    ))

## Evaluate prediction based on model fit
xtest <- seq(from=0, by=0.001,to=1)
pred <- predict(fit,as.matrix(xtest),predictAll=T)
ypred <- pred$f
spred <- pred$s

## Plot the prediction of the model:
curve(objectFun(x),0,1)
points(x,y)
lines(xtest,ypred,lty=2,lwd=2)

## Plot suggested candidate solution
points(xtest[which.min(ypred)],ypred[which.min(ypred)],col="black",pch=20,cex=2)
```
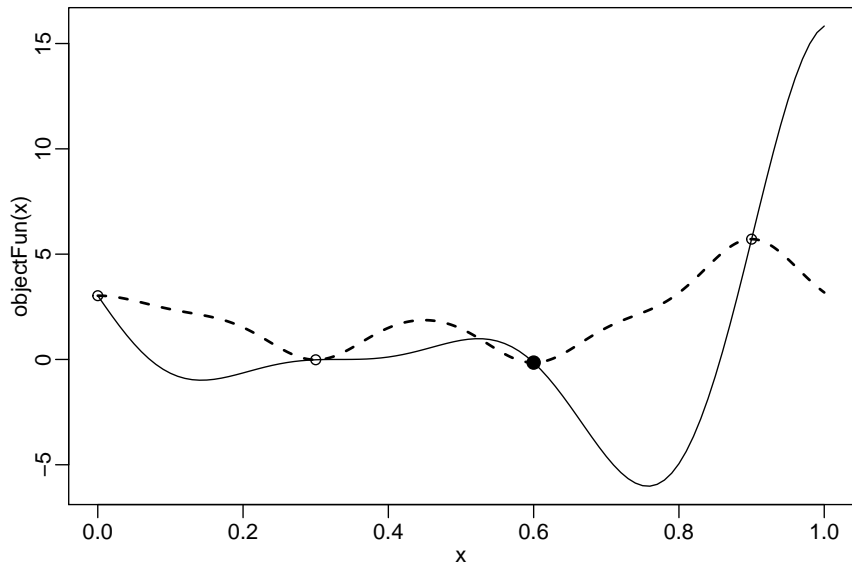
```
## Calculate expected improvement (EI)
ei <- 10^(-spotInfillExpImp(ypred,spred,min(y)))
## note: the function used above returns negative
## log. of EI, for optimization purposes.

## Plot EI
curve(objectFun(x),0,1)
points(x,y)
lines(xtest,ypred,lty=2,lwd=2)
par(new = T)
plot(xtest,ei,lty=3,lwd=2, type="l", axes=F, xlab=NA, ylab=NA,
     ylim=rev(range(ei)))
axis(side = 4); mtext(side = 4, line = 1.4, 'EI')

## Determine solution that maximizes EI
newx <- xtest[which.max(ei)]

## Plot suggested candidate solution, based on EI
points(newx,max(ei),col="red",pch=20,cex=2)

## Add data
x <- c(x,newx)
y <- c(y,objectFun(newx))
```
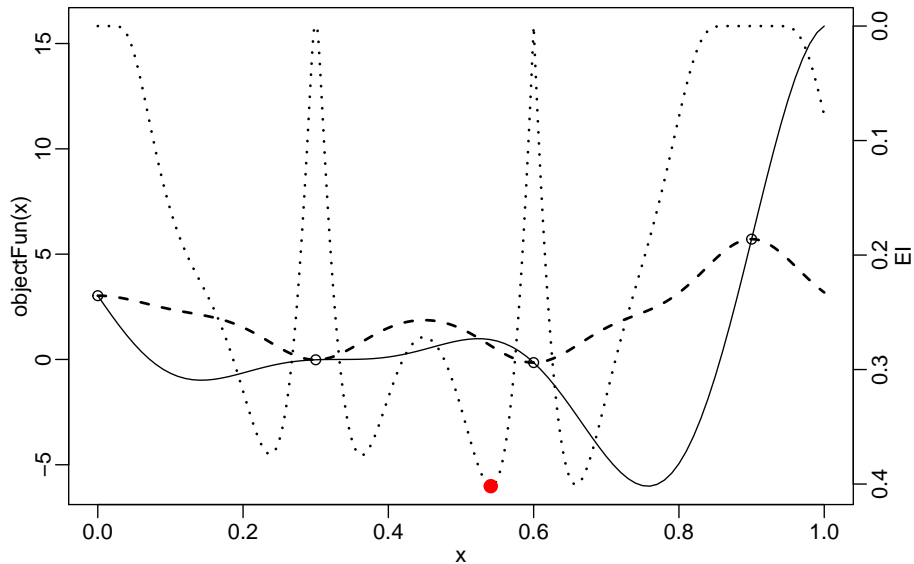
```r
## Now repeat the same as often as necessary:
repeatThis <- expression({
  curve(objectFun(x),0,1)
  points(x,y)
  fit <- forrBuilder(as.matrix(x),as.matrix(y),
                     control=list(uselambda=FALSE
                     ))
  xtest <- seq(from=0, by=0.001,to=1)
  pred <- predict(fit,as.matrix(xtest),predictAll=T)
  ypred <- pred$f
  spred <- pred$s
  lines(xtest,ypred,lty=2,lwd=2)
  points(xtest[which.min(ypred)],ypred[which.min(ypred)],col="black",pch=20,cex=2)
  ei <- 10^(-spotInfillExpImp(ypred,spred,min(y)))
  par(new = T)
  plot(xtest,ei,lty=3,lwd=2, type="l", axes=F, xlab=NA, ylab=NA,
           ylim=rev(range(ei)))
  axis(side = 4); mtext(side = 4, line = 1.4, 'EI')
  points(xtest[which.max(ei)],max(ei),col="red",pch=20,cex=2)
  newx <- xtest[which.max(ei)]
  x <- c(x,newx)
  y <- c(y,objectFun(newx))
})
```

```
eval(repeatThis)
```
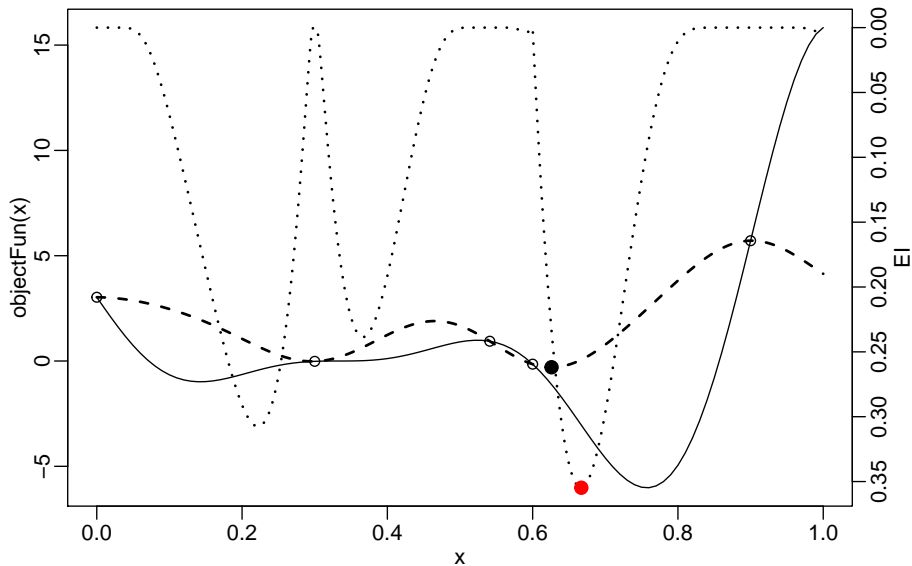
```
eval(repeatThis)
```
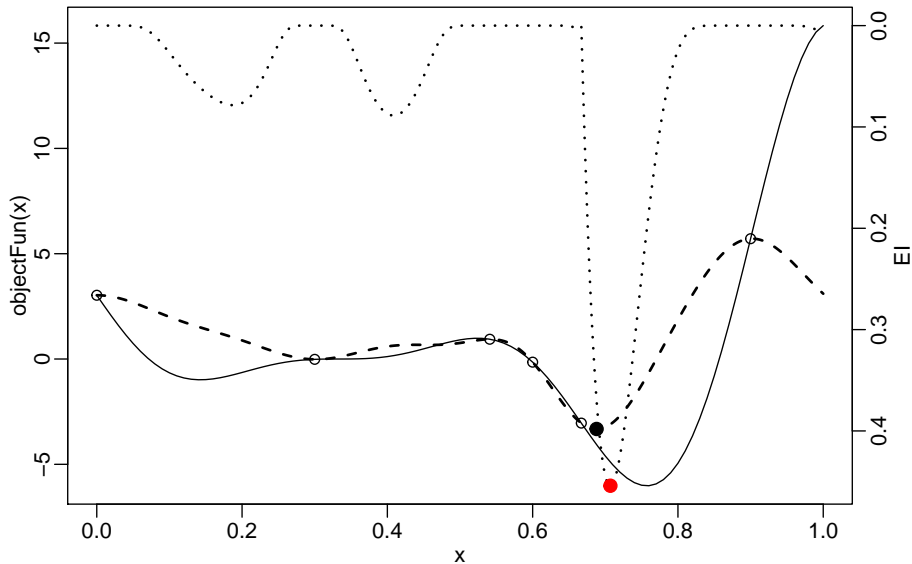
```
eval(repeatThis)
```
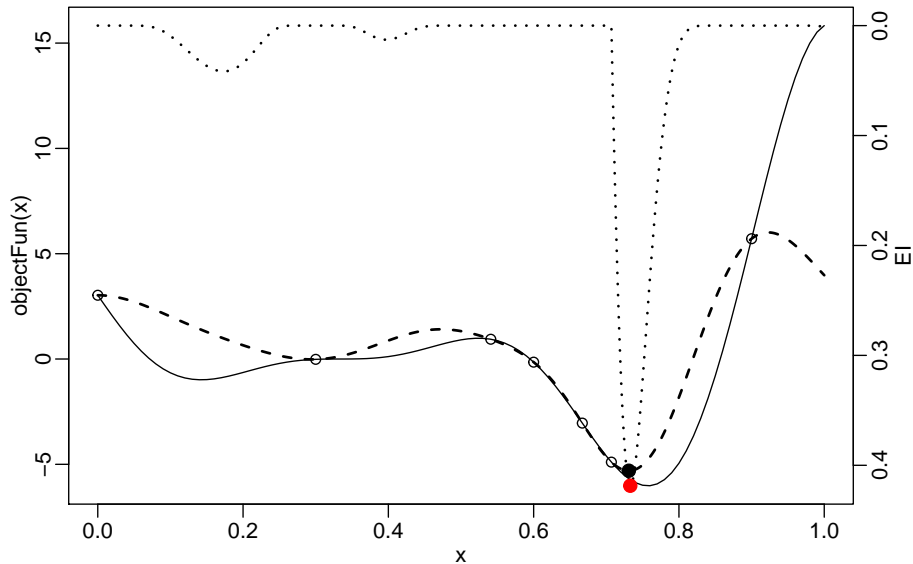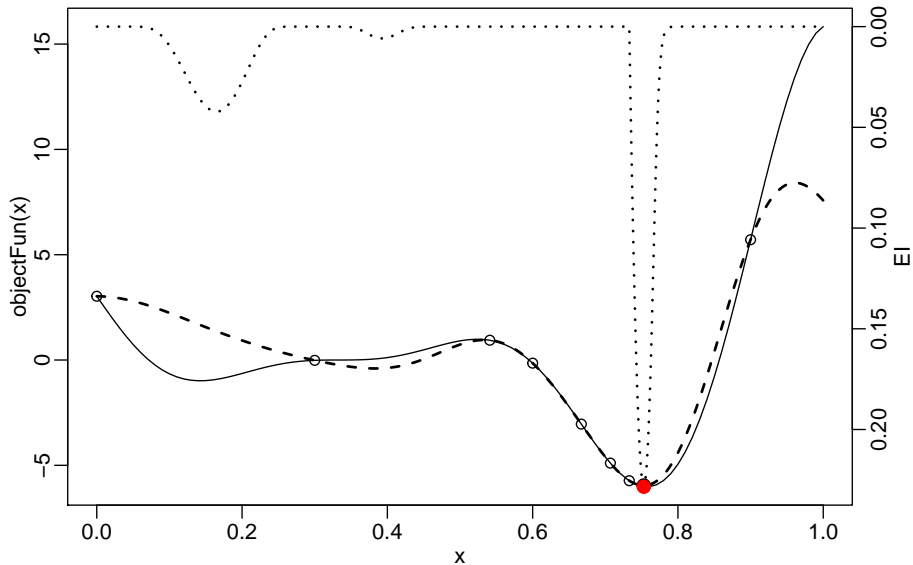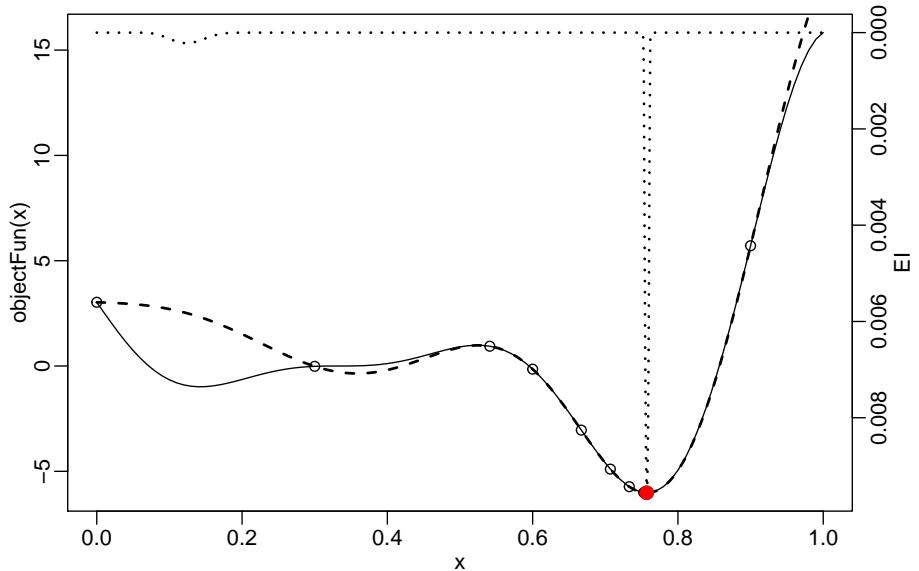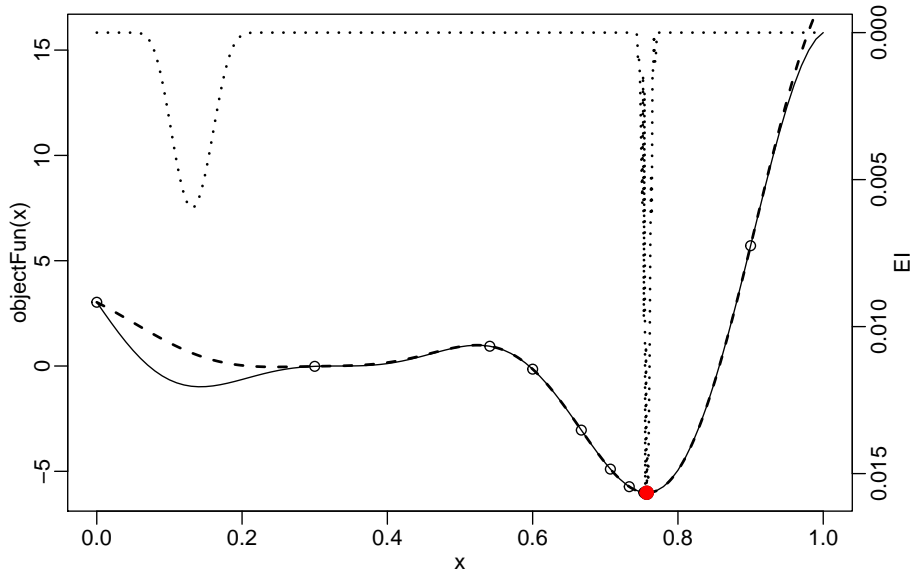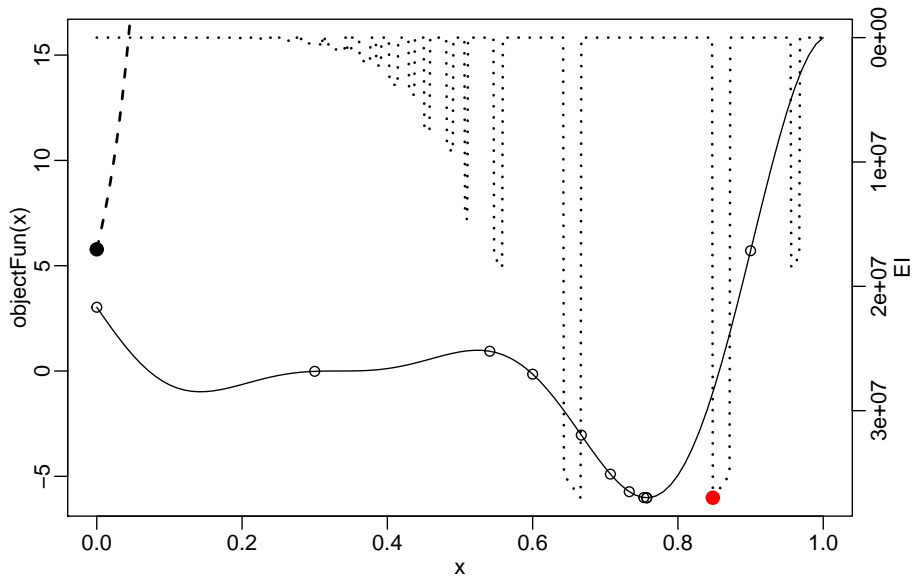
```
eval(repeatThis)
```
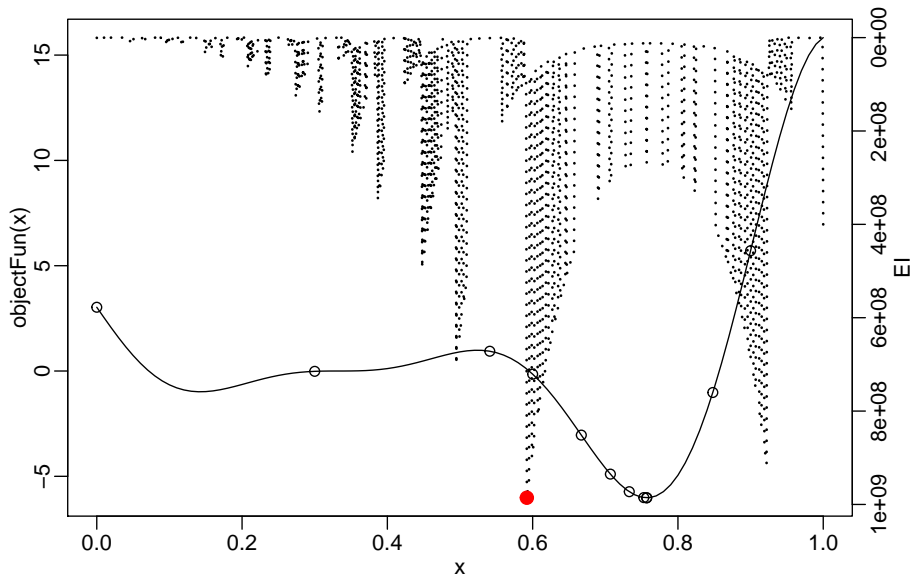
- EI looks noisy, strange
- Predicted mean is completely off target
- Why?

- Common practical problem
  - Numerical issue
  - Closeness of solutions
  - Problem for Kriging model
    - Near identical rows in correlation matrix
    - Badly conditioned
  - Often, Kriging implementations will crash
  - Here, nonsensical predictions

- Potential remedy: use nugget (regularization) + reinterpolation
  [Forrester et al. 2008]

```r
## repeat as often as necessary (but now with regularization):
repeatThis <- expression({
  curve(objectFun(x),0,1)
  points(x,y)
  fit <- forrBuilder(as.matrix(x),as.matrix(y),
            control=list(
            uselambda=TRUE, # Use nugget (parameter lambda)
            reinterpolate=T # Reinterpolation, to fix uncertainty estimates
                ))
  xtest <- seq(from=0, by=0.001,to=1)
  pred <- predict(fit,as.matrix(xtest),predictAll=T)
  ypred <- pred$f
  spred <- pred$s
  lines(xtest,ypred,lty=2,lwd=2)
  points(xtest[which.min(ypred)],ypred[which.min(ypred)],col="black",pch=20,cex=2)
  ei <- 10^(-spotInfillExpImp(ypred,spred,min(y)))
  par(new = T)
  plot(xtest,ei,lty=3,lwd=2, type="l", axes=F, xlab=NA, ylab=NA,
            ylim=rev(range(ei)))
  axis(side = 4); mtext(side = 4, line = 1.4, 'EI')
  points(xtest[which.max(ei)],max(ei),col="red",pch=20,cex=2)
  newx <- xtest[which.max(ei)]
  x <- c(x,newx)
  y <- c(y,objectFun(newx))
})
```

```r
eval(repeatThis)
```

Overview

- Motivation

- Concepts and methods

- Practical approach: instructive application

- **Typical problems in application**

- Open Issues / Research perspectives / Fields of Interest
    - Multi-criteria optimization
    - Combinatorial optimization

- Discussion
    - Typical problems and their solutions

Typical Problems in Practice

- Previous slides: numerical issues (Kriging)

- Other, more general issues:
  - Problem definition
    - What is the objective
    - What variables impact the objective
    - ...
  - Algorithm design, selection of:
    - Model
    - Optimizer
    - Parameters

Typical Problems in Practice: Problem definition

- Very important, crucial to success
- Often underestimated
- Information based on
  - **Discussions with application experts, practitioners**
  - Literature
  - Experience

Typical Problems in Practice: Problem definition

- Consider the following:
  - Aims and goals
    - What are they?
    - Can they be clearly defined?
    - Can they be evaluated (measured, computed)?
    - Cost of evaluation?
    - Budget?
    - Desired accuracy?
  - Variables affecting the objective(s)
    - How many?
    - Independent variables?
    - Disturbance variables?
    - Data types?
  - Constraints?
  - Noise?
  - Interfacing, data exchange

- Repeat the aforementioned, e.g., after first results
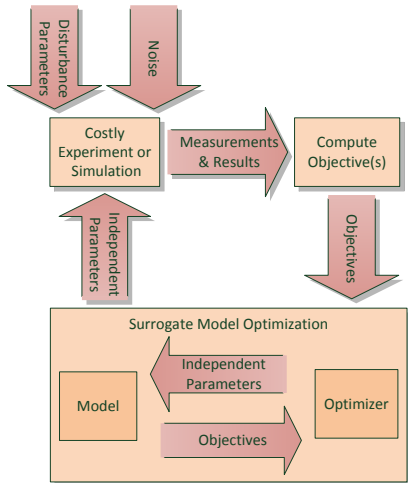
Typical Problems in Practice: Model Selection
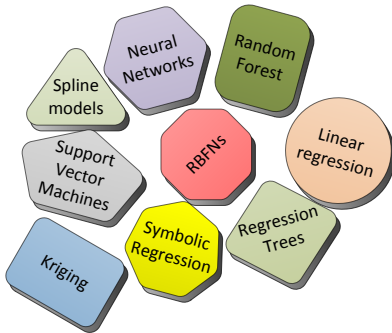
- Large variety of models available
- Which to choose?

- Potential solutions:
  - Use the "default" (e.g., Kriging / EGO)
  - Exploit problem knowledge
  - Select performance-based or combine ->
    Ensembles (open issue)

Typical Problems in Practice: Model Selection

- No problem is truly black-box
- Use what you know, e.g.:

- Number of parameters
  - 20 or more: Kriging and related loose performance
- Data types
  - Continuous: Kriging, SVMs, RBFNs
  - Integer, binary, categorical parameters: e.g., Random Forest
  - Mixed: Treed Gaussian Processes (TGP)
  - Structured / combinatorial (e.g.: permutations, trees): see later slides
- Data set sizes (budget)
  - Large: Kriging may become slow
  - Small: Take care to use models that avoid overfitting
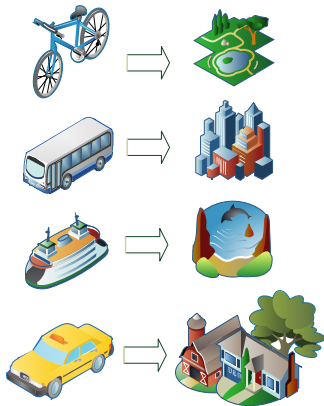
Typical Problems in Practice: Model Selection

- Structure of the fitness landscape:
  - Highly multi-modal: do not use simple linear models
  - Smooth: Kriging or related
  - Large plateaus or discontinuities: Kriging variants may perform poorly
  - Known trend: Use Kriging with trend function.
- Cost of the objective function
  - Rather high: Complex, powerful models (Kriging, SVMs)
  - Rather low: Less complex, cheaper models (linear regression, tree-based, k-Nearest Neighbor)
- Requirements of understandability / learning from the model
  - Variable importance: most models
  - Rule extraction: regression trees
  - Human readable formulas: linear models, genetic programming (symbolic regression)
- Availability of derivatives
  - e.g., Gradient Enhanced Kriging **?**
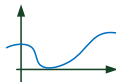
Typical Problems in Practice: Model Selection

- Other considerations
  - Customer / Practitioner preferences and knowledge
    - Do they understand the models
    - Do they trust results from the models

  - Your own preferences & experience
    - e.g., with regards to parameterization
    - or implementation

  - Note, various model types often quite similar & related, interchangeable
  - e.g.: Spline models - Kriging - SVM - RBFN

Typical Problems in Practice: Implementation

- Once models are selected -> Implementation
- Can have significant impact
- Options
  - Frequently employed packages/libraries
    - Quality
    - Community support
    - Examples, documentation
    - Continuity of development
  - Less frequently used work
    - For special tasks?
    - Because of specific features?
  - Do it yourself
    - None other available (or too slow, buggy)
    - Specific features not available
    - More fun ☺ but also more work ☹
    - **You know what the model really does**

Typical Problems in Practice: Optimizer Selection

- Similar considerations as for models

- Optimizer also depends on model type (and vice versa)

  - Smooth, differentiable models like Kriging: gradient-based optimizers are fine

  - Non-smooth (tree-based): GA, DE, PSO

  - Multimodality (of prediction or infill criterion, e.g., EI):
    Population based, restarts, niching, etc.

  - Simple linear regression: analytical

Typical Problems in Practice: Parameter Selection

- Similar to model selection / optimizer selection ...
- ... but with more attention to details
  - Use expert / literature suggestions
  - Exploit problem knowledge
    - Parameters affect:
    - complexity,
    - cost of modeling,
    - cost of model optimization,
    - noise handling,
    - robustness,
    - smoothness,
    - ...
  - Tuning, benchmarks (open issue)

Overview

- Motivation

- Concepts and methods

- Practical approach: instructive application

- Typical problems in application

- **Open Issues / Research perspectives / Fields of Interest**
  - Multi-criteria optimization
  - Combinatorial optimization

- Discussion
  - Typical problems and their solutions

Open Issues

- Research perspectives
- Fields of Interest

    - Multi-objective:
      SAMCO - Surrogate Assisted Multi-Criteria Optimisation
    - Combinatorial surrogates models (optimisation)

Open Issues

- Research perspectives
- Fields of Interest

    - Multi-objective:
      SAMCO - Surrogate Assisted Multi-Criteria Optimisation
    - Combinatorial surrogates models (optimisation)
    - … both handled in more detail later!

Open Issues

- Meaningful benchmarking and testing of algorithms
- Noise handling
- Complex resource limitations
- High-dimensional / large scale data
- Constraint handling
- Aggregation: Model ensembles, Multi-fidelity models
- Dynamic optimization problems

Open Issues

- Meaningful benchmarking and testing of algorithms
    - Some benchmark sets available
    - (Almost?) not considered for evaluation
    - No standard implemented
    - Depending on people who apply?

Open Issues

- Meaningful benchmarking and testing of algorithms
    - Some benchmark sets available
    - (Almost?) not considered for evaluation
    - No standard implemented
    - Depending on people who apply?

- Noise handling
    - Surrogates considered for noisy problems
    - What about noise in models?

Open Issues

- Meaningful benchmarking and testing of algorithms
  - Some benchmark sets available
  - (Almost?) not considered for evaluation
  - No standard implemented
  - Depending on people who apply?

- Noise handling
  - Surrogates considered for noisy problems
  - What about noise in models?

- Complex resource limitations
  - Resources like computation times may not be available constantly
  - Server availability, different calculation times per job ...
  - Problem handled separately
  - Integration of resources handling in algorithm needed

Open Issues

- High-dimensional / large scale data
  - Models may fail / not be applicable
  - New models might need to be considered
  - New integration schemes needed as well?

Open Issues

- High-dimensional / large scale data
  - Models may fail / not be applicable
  - New models might need to be considered
  - New integration schemes needed as well?

- Constraint handling
  - Different scenarios possible
  - Most common: infeasible offspring of feasible ancestor
    - Easy strategy: just omit

Open Issues

- High-dimensional / large scale data
  - Models may fail / not be applicable
  - New models might need to be considered
  - New integration schemes needed as well?

- Constraint handling
  - Different scenarios possible
  - Most common: infeasible offspring of feasible ancestor
    - Easy strategy: just omit . . . optimal?
  - Constraints to be considered by models as well?
  - Integration in algorithms?
  - Optimal strategy?

Open Issues

- Aggregation: Model ensembles, Multi-fidelity models
  - Which model in which situation?
    - Again depending on many parameters
    - Some results available . . .
  - How to aggregate ensembles best?
  - Setting may vary over time . . .

Open Issues

- Aggregation: Model ensembles, Multi-fidelity models
  - Which model in which situation?
    - Again depending on many parameters
    - Some results available . . .
  - How to aggregate ensembles best?
  - Setting may vary over time . . .

- Dynamic optimization problems
  - In general: time-varying fitness function
  - Surrogates used for forecasting, predicting future values
  - Other settings possible . . . see above

Overview

- Motivation

- Concepts and methods

- Practical approach: instructive application

- Typical problems in application

- Open Issues / Research perspectives / Fields of Interest
  - **Multi-criteria optimization**
  - Combinatorial optimization

- Discussion
  - Typical problems and their solutions

SAMCO

Surrogate-Assisted Multi-Criteria Optimisation

- Intersection of
  - Multi-criteria optimisation
  - Surrogate-assisted optimisation

- MCO – Multi-Criteria Optimisation
- EMO – Evolutionary Multi-objective Optimisation
- EMOA – Evolutionary Multi-objective Optimisation Algorithm
- MOEA – Multi-Objective Evolutionary Optimisation

Basics of Multi-criteria optimisation

- Multiple objective functions considered

- Minimize

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}^m, \quad f(x) = (f_1(x), \ldots, f_m(x))$$

Pareto Dominance

- Solution x dominates solution y

$$
\begin{aligned}
x <_p y \quad :\Leftrightarrow \quad & \forall i : \quad f_i(x) \leq f_i(y) && (i = 1, \ldots m) \\
& \exists j : \quad f_j(x) < f_j(y) && (j = 1, \ldots m)
\end{aligned}
$$

- **Pareto-Set**: Set of all non-dominated solutions in the search space

$$\{x \mid \nexists z : \quad z <_p x\}$$

- **Pareto-Front**: Image of Pareto-set in objective space

SAMCO

- Available budget: 100 to $10\,000$ evaluations
- Different strategies
    - Stochastic variation of EAs assisted (e.g. filtering solutions)
    - Completely replaced (e.g. optimizing figure of merit)
- Many algorithms already developed
- However
    - Very heterogeneous research fields
    - Different sciences / faculties involved
        - Engineering
        - Statistics
        - Computer Science
        - Mathematics
        - Aeronautics
        - Agriculture
- **Thus:** different backgrounds, also different languages to be considered

SAMCO

- Application driven
  - Proposed algorithms tested respective application tasks mainly
  - Comparison of different approaches hard to accomplish
  - Lacks existence of accepted benchmarks

- Theoretical aspects almost neglected due to focus on practical applications

- Methodological research areas
  - Choice of the surrogate model
  - Respective figure of merit (or infill criterion)

SAMCO

- Easiest approach: one model per objective
- Kriging and expected improvement used commonly

- 🔴 Precise Evaluationson the old Pareto front
- 🔵 Mean values of approximations
- ⚪ Lower confidence bounds
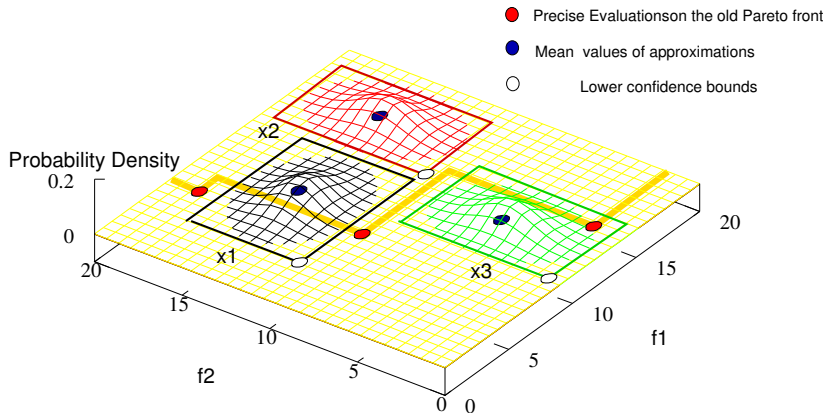
Image taken from [Emmerich et al., 2006]

Alternative approaches

ParEGO – Pareto Efficient Global Optimisation [Knowles, 2006]

- Converts different cost values into single one
  - Parameterized scalarizing weight vector (augmented Tchebycheff function)
  - Using augmented Tchebycheff function
  - Different weight vector at each iteration
  - Weight vector is drawn uniformly at random
  - Allows for gradually building an approximation to whole Pareto front

- Learns a Gaussian processes model of search landscape
  - Scalar costs of all previously visited solutions is computed
  - DACE model of landscape is constructed by maximum-likelihood
  - Solution that maximizes expected improvement becomes next point
  - Evaluation on real, expensive cost function
  - Update after every function evaluation

- Ensures that weakly dominated solutions are rewarded less than Pareto optimal ones

Alternative approaches

RASM – Rank-based aggregated surrogate models [Loshchilov et al., 2010]

- Mono-surrogate approach again

- Single surrogate model to reflect Pareto dominance in EMO framework

- Locally approximates Pareto dominance relation
  - Ranking neighbor points within the objective space
  - Offspring filter estimating whether they improve on their parents in terms of approximated Pareto-dominance
  - Used for offspring generation in standard EMOA

- Modeling Pareto dominance within the rank-SVM framework

Existing libraries and approaches

- Many libraries already existing, e.g.

    - mlrMBO
    - DiceKriging
    - SUMO
    - parEGO

    - GPareto
    - SPOT
    - Shark
    - QstatLab

  - Overview on SAMCO homepage:
    `http://samco.gforge.inria.fr/doku.php?id=surr_mco`

- **However:** up-to-date overview is missing
  - List algorithms contained
  - Compare strengths and weaknesses

SAMCO Promising research areas

- Multiple objectives with different response surfaces
  + specific requirements of set- and indicator- based optimization

- New variants of models

- New infill criteria

SAMCO Promising research areas

- Multiple objectives with different response surfaces
  + specific requirements of set- and indicator- based optimization

- New variants of models

- New infill criteria

- Approaches beyond one model per objective function
  - Model dominance relations
  - Model performance indicator landscapes

- Ensembles of Surrogates
  - Multiple surrogates simultaneously or successively
    - To improve overall quality of prediction of each objective
    - Model evolves over time from a coarse grained to finer one
    - Different parts of search space with significantly different behavior

SAMCO Promising research areas

- Collect existing approaches and libraries

- Benchmarking Surrogate-Assisted Optimizers lacks rigorously
  - Review of common test functions (academic vs. real-world)
  - Understand weaknesses and strengths of each algorithm
  - Algorithm recommendations for practice

  - Overview on SAMCO homepage:
    http://samco.gforge.inria.fr/doku.php?id=benchmarking

Overview

- Motivation

- Concepts and methods

- Practical approach: instructive application

- Typical problems in application

- Open Issues / Research perspectives / Fields of Interest
  - Multi-criteria optimization
  - **Combinatorial optimization**

- Discussion
  - Typical problems and their solutions

Discrete / combinatorial / structured search spaces

- Well established in expensive, continuous optimization

What about combinatorial / discrete optimization problems?

- Let's get an overview

Survey: combinatorial surrogates

Mixed variables

model	optimizer	cost	budget	dimension	remarks / topics	reference
RBFN	ES	cheap / $\sim$ expensive	560 / 280	15 / 23	benchmark / real-world: medical image analysis	Li et al. [2008]
Random Forest, Kriging	NSGA2	$\sim$expensive	-	4-76	algorithm tuning	Hutter et al. [2010]
RBFN + cluster	GA	cheap	2,000	12	benchmark, real-world: chemical industry	Bajer and Holeňa [2010]
RBFN + GLM	GA	cheap	several thousand	4-13	benchmark, real-world: chemical industry	Bajer and Holeňa [2013]
SVR	NSGA2	?	2,000	10	finite element, multi criteria	Herrera et al. [2014]

Binary strings

model	optimizer	cost	budget	dimension	remarks / topics	reference
ANN	SA	expensive	?	16	real world, pump positioning	Rao and Manju [2007]
RBFN	GA	cheap	dimension^2	10-25	NK-Landscape	Moraglio and Kattan [2011a]
RBFN	GA	expensive	100	10-40	benchmark, package deal negotiation	Fatima and Kattan [2011]
Kriging	GA	cheap	dimension^2	10-25	NK-Landscape	Zaefferer et al. [2014b]

Survey: combinatorial surrogates

Permutations

model	optimizer	cost	budget	dimension	remarks / topics	reference
custom	brute force	expensive	28	6	signed permutation, real world: weld sequence	Voutchkov et al. [2005]
RBFN	GA	cheap	100	30 - 32	benchmark	Moraglio et al. [2011]
Kriging	GA	cheap	100	12 - 32	benchmark	Zaefferer et al. [2014b]
Kriging	GA	cheap	200	10 - 50	distance selection	Zaefferer et al. [2014a]
Kriging	ACO	cheap	100 - 1,000	50 - 100	benchmark, tuning	Pérez Cáceres et al. [2015]
RBFN	GA*	instance dependent	1,000	50 - 1,928	numerical stability, real world: cell suppression	Smith et al. [2016]
Kriging	brute force, GA	cheap	100	5 - 10	kernel definiteness	Zaefferer and Bartz-Beielstein [2016]

*Different integration: GA produces random solutions, which are filtered by the model in each iteration

Survey: combinatorial surrogates

Trees

model	optimizer	cost	budget	remarks / topics	reference
RBFN	GA	cheap	100	symbolic regression	Moraglio and Kattan [2011b]
kNN	GA	expensive	30,000	Phenotypic similarity, genetic programming	Hildebrandt and Branke [2014]
RBFN*	GA	cheap	100	symbolic regression, parity	Kattan and Ong [2015]
Random Forest	GA	cheap	15,000	benchmark, genetic programming	Pilát and Neruda [2016]

*two models: semantic and fitness

Other

model	optimizer	cost	budget	dimension	remarks / topics	reference
k-NN	GA	rather cheap	20000 - 200000	161 - 259	real-valued+structure, real-world, protein structure	Custódio et al. [2010]
Kriging	GA	expensive	few hundreds		graph-based, real-world, protein structure	Romero et al. [2013]
ANN	DE	cheap	several hundreds	40 - 500	assignment problem, dynamic	Hao et al. [2016]

Summary: Strategies

- Strategies of dealing with discrete / combinatorial search spaces
  - Inherently discrete models (e.g., regression trees)
    - simple, but may not be efficient/feasible for any representation
  - Dummy variables
    - Only for linear regression, vector-based
  - Feature based
    - Extract real-valued features from genotype / phenotype
    - Requires good features
  - (Dis)similarity measure based (distance, kernel)
    - Requires good measure

Summary: Types of Models

- As varied as in the continuous case:

  - Custom, application specific models (expert knowledge, physics)

  - Artificial Neural Networks (ANN)

  - Markov Random Fields [Allmendinger et al., 2015]

  - Random Forest (Integer, Mixed Integer Problems)

  - (Probabilistic models - in Estimation of Distribution Algorithms)

  - (Pheromone trails - in Ant Colony Optimization)

  - "Classical" kernel-based (similarity-based) models:
    - k-Nearest Neighbour (k-NN)
    - Radial Basis Function Networks (RBFN)
    - Support Vector Regression (SVR)
    - **Kriging (Gaussian Processes)**
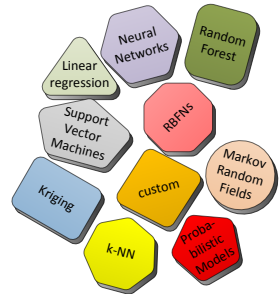
Why kernel based approach, Kriging?

- Conceptually simple:
    - Replace kernel or distance function
    - e.g., with Gaussian kernel and arbitrary distance:

$$k(x, x') = \exp(-\theta \mathsf{d}(x, x'))$$

- Transfer of popular method from continuous domain
    - Powerful predictor
    - Elegant parameter fitting (maximum likelihood estimation)
    - Uncertainty estimate, Expected Improvement
    - $\rightarrow$ Efficient Global Optimization EGO [Jones et al., 1998b]

- Note:
    - None of these features exclusive to Kriging
    - Closely related to other model types

# Combinatorial Surrogates: Research Questions

- Which kernel/distance works best and why?

- **How to choose a suitable kernel/distance?**

- Or else, combine?

- Genotypic vs phenotypic distances? [Hildebrandt and Branke, 2014]

- **Definiteness?**

- Dimensionality issues? Dimensionality reduction? See e.g., the very high dimensional problems in [Smith et al., 2016]

- Comparison of model types?[1]

- And again: benchmarking / testing?

---

[1]If you want to compare your approach to our methods: R Package for Combinatorial Efficient Global Optimization CEGO - `https://cran.r-project.org/package=CEGO`.

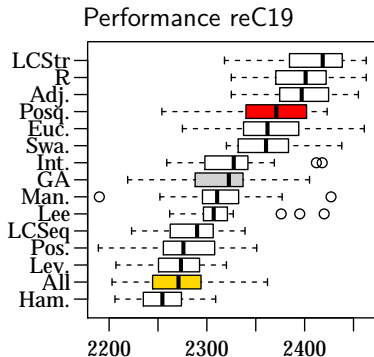# Research Question: Choosing a Distance / Kernel

[Zaefferer et al., 2014a]

* Choice crucial for success
* Use prior knowledge (if available?)
* Cross-validation

- Fitness Distance Correlation (FDC) (potentially misleading)

FDC
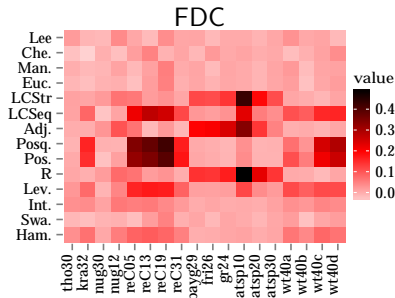
- Maximum Lilkelihood Estimation (MLE) (seems to work well)

Performance reC19

Research Question: Definiteness

[Zaefferer and Bartz-Beielstein, 2016]

So we can just replace the distance or kernel function with something appropriate and everything is fine, right?

Common requirement for kernels (distances): Definiteness[2]

- Definiteness may be unknown / lacking
- Designing definite kernels may be hard / infeasible
- Required: correction procedure
- Some results from SVM field [Ong et al., 2004; Chen et al., 2009; Loosli et al., 2015] Survey: [Schleif and Tino, 2015]
- Can be transfered to Kriging, with some tweaks

---

[2]Positive semi-definite kernel matrix: all eigenvalues are positive or zero

That's all Folks. Thanks for hanging on.

- Any questions?

- Discussion:

    - Problems you encountered in practice?

    - New directions, challenges?

    - What is missing in the field?

    - Interesting applications?

Allmendinger, R., Coello, C. A. C., Emmerich, M. T. M., Hakanen, J., Jin, Y., and Rigoni, E. (2015). Surrogate-assisted multicriteria optimization (wg6). In Greco, S., Klamroth, K., Knowles, J. D., and Rudolph, G., editors, *Understanding Complexity in Multiobjective Optimization (Dagstuhl Seminar 15031) - Dagstuhl Reports*, volume 5, pages 96–163, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Bajer, L. and Holeňa, M. (2010). Surrogate model for continuous and discrete genetic optimization based on rbf networks. In *Intelligent Data Engineering and Automated Learning – IDEAL 2010*, volume 6283 LNCS, pages 251–258.

Bajer, L. and Holeňa, M. (2013). Surrogate model for mixed-variables evolutionary optimization based on glm and rbf networks. In *SOFSEM 2013: Theory and Practice of Computer Science*, volume 7741 LNCS, pages 481–490.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Chen, Y., Gupta, M. R., and Recht, B. (2009). Learning kernels from indefinite similarities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 145–152, New York, NY, USA. ACM.

Custódio, F. L., Barbosa, H. J., and Dardenne, L. E. (2010). Full-atom ab initio protein structure prediction with a genetic algorithm using a similarity-based surrogate model. In *Proceedings of the Congress on Evolutionary Computation (CEC'10)*, pages 1–8, New York, NY, USA. IEEE.

Deng, L. and Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387.

Emmerich, M. T. M., Giannakoglou, K. C., and Naujoks, B. (2006). Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439.

Fatima, S. and Kattan, A. (2011). Evolving optimal agendas for package deal negotiation. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 505–512, New York, NY, USA. ACM.

Flasch, O., Mersmann, O., and Bartz-Beielstein, T. (2010). Rgp: An open source genetic programming system for the r environment. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '10, pages 2071–2072, New York, NY, USA. ACM.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering Design via Surrogate Modelling*. Wiley.

Forrester, A. I. and Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79.

Hao, J., Liu, M., Lin, J., and Wu, C. (2016). A hybrid differential evolution approach based on surrogate modelling for scheduling bottleneck stages. *Computers & Operations Research*, 66:215–224.

Haykin, S. (2004). A comprehensive foundation. *Neural Networks*, 2(2004).

Herrera, M., Guglielmetti, A., Xiao, M., and Filomeno Coelho, R. (2014). Metamodel-assisted optimization based on multiple kernel regression for mixed variables. *Structural and Multidisciplinary Optimization*, 49(6):979–991.

Hildebrandt, T. and Branke, J. (2014). On using surrogates with genetic programming. *Evolutionary Computation*, pages 1–25.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2010). Sequential model-based optimization for general algorithm configuration (extended version). Technical Report TR-2010-10, University of British Columbia, Department of Computer Science. Available online: http://www.cs.ubc.ca/~hutter/papers/10-TR-SMAC.pdf.

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998a). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998b). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

Kattan, A. and Ong, Y.-S. (2015). Surrogate genetic programming: A semantic aware evolutionary search. *Information Sciences*, 296:345–359.

Knowles, J. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

Li, R., Emmerich, M. T. M., Eggermont, J., Bovenkamp, E. G. P., Bäck, T., Dijkstra, J., and Reiber, J. (2008). Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In *Proceedings of the Congress on Evolutionary Computation (CEC'08)*, pages 2764–2771, New York, NY, USA. IEEE.

Loosli, G., Canu, S., and Ong, C. (2015). Learning svm in krein spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1204–1216.

Loshchilov, I., Schoenauer, M., and Sebag, M. (2010). Dominance-Based Pareto-Surrogate for Multi-Objective Optimization. In *Simulated Evolution and Learning (SEAL 2010)*, volume 6457 of *LNCS*, pages 230–239. Springer.

Moraglio, A. and Kattan, A. (2011a). Geometric generalisation of surrogate model based optimisation to combinatorial spaces. In *Proceedings of the 11th European Conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP'11, pages 142–154, Berlin, Heidelberg, Germany. Springer.

Moraglio, A. and Kattan, A. (2011b). Geometric surrogate model based optimisation for genetic programming: Initial experiments. Technical report, University of Birmingham.

Moraglio, A., Kim, Y.-H., and Yoon, Y. (2011). Geometric surrogate-based optimisation for permutation-based problems. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '11, pages 133–134, New York, NY, USA. ACM.

Naujoks, B., Steden, M., Muller, S. B., and Hundemer, J. (2007). Evolutionary optimization of ship propulsion systems. In *2007 IEEE Congress on Evolutionary Computation*, pages 2809–2816.

Ong, C. S., Mary, X., Canu, S., and Smola, A. J. (2004). Learning with non-positive kernels. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 81–88, New York, NY, USA. ACM.

Pilát, M. and Neruda, R. (2016). Feature extraction for surrogate models in genetic programming. In *Parallel Problem Solving from Nature – PPSN XIV*, pages 335–344. Springer Nature.

Pérez Cáceres, L., López-Ibáñez, M., and Stützle, T. (2015). Ant colony optimization on a limited budget of evaluations. *Swarm Intelligence*, pages 1–22.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K. (2005). Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28.

Rao, S. V. N. and Manju, S. (2007). Optimal pumping locations of skimming wells. *Hydrological Sciences Journal*, 52(2):352–361.

Romero, P. A., Krause, A., and Arnold, F. H. (2013). Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, pages 409–423.

Schleif, F.-M. and Tino, P. (2015). Indefinite proximity learning: A review. *Neural Computation*, 27(10):2039–2096.

Smith, J., Stone, C., and Serpell, M. (2016). Exploiting diverse distance metrics for surrogate-based optimisation of ordering problems: A case study. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, GECCO '16, pages 701–708, New York, NY, USA. ACM.

Voutchkov, I., Keane, A., Bhaskar, A., and Olsen, T. M. (2005). Weld sequence optimization: The use of surrogate models for solving sequential combinatorial problems. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33):3535–3551.

Zaefferer, M. and Bartz-Beielstein, T. (2016). Efficient global optimization with indefinite kernels. In *Parallel Problem Solving from Nature–PPSN XIV*, pages 69–79. Springer.

Zaefferer, M., Stork, J., and Bartz-Beielstein, T. (2014a). Distance measures for permutations in combinatorial efficient global optimization. In Bartz-Beielstein, T., Branke, J., Filipic, B., and Smith, J., editors, *Parallel Problem Solving from Nature–PPSN XIII*, pages 373–383, Cham, Switzerland. Springer.

Zaefferer, M., Stork, J., Friese, M., Fischbach, A., Naujoks, B., and Bartz-Beielstein, T. (2014b). Efficient global optimization for combinatorial problems. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 871–878, New York, NY, USA. ACM.