Sequential Parameter Optimization

T. Bartz-Beielstein

Department of Computer Science
Cologne University of Applied Sciences

April 28, 2009

Outline

Goals

- Understanding how algorithms work
- Improve performance
- Tailor algorithms to problems
- Provide software, which enables a fair and statstically sound comparison
- Two topics:
  - Sequential parameter optimization (SPO): General framwork
  - Sequential parameter optimization toolbox (SPOT): one implementation of this framework. Available in matlab and R

Applications 1/2

SPOT was successfully applied to numerous optimization algorithms, especially in the field of evolutionary computation, i.e., evolution strategies, particle swarm optimization, genetic programming etc. in the following domains:

- **machine engineering**: design of mold temperature control (Mehnen et al., 2005; Weinert et al., 2004; Mehnen et al., 2004)
- **aerospace industry**: airfoil design optimization (Bartz-Beielstein and Naujoks, 2004)
- **simulation and optimization**: elevator group control (Bartz-Beielstein et al., 2005c; Markon et al., 2006)
- **technical thermodynamics**: non sharp separation (Bartz-Beielstein et al., 2005b)
- **economy**: agri-environmental policy-switchings (de Vegt, 2005)
- **logistics**: vehicle routing and door-assignment problems (Bartz-Beielstein et al., 2006)

Applications 2/2

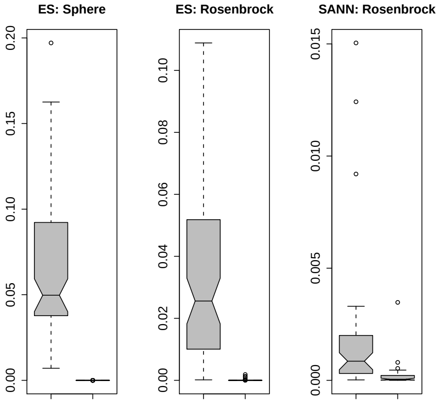
Other fields of application are in fundamental research:

- **algorithm engineering**: graph drawing (Tosic, 2006)
- **statistics**: selection under uncertainty (optimal computational budget allocation) for PSO (Bartz-Beielstein et al., 2005a)
- **evolution strategies**: threshold selection and step-size adaptation (Bartz-Beielstein, 2005)
- **other evolutionary algorithms**: genetic chromodynamics (Stoean et al., 2005)
- **computational intelligence**: algorithmic chemistry (Bartz-Beielstein et al., 2005b; Lasarczyk, 2007)
- **particle swarm optimization**: analysis and application (Bartz-Beielstein et al., 2004a)
- **numerics**: comparison and analysis of classical and modern optimization algorithms (Bartz-Beielstein et al., 2004c)

Further projects (waste-water treatment) are subject of current research.

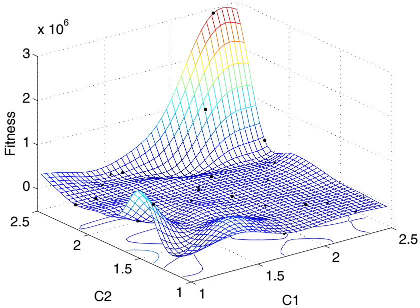
Starting Point: Classical DoE

- Inspired by Kleijnen (1987)
- DoE applied to
  a) stochastic search, e.g., evolution strategies
  b) deterministic search, e.g., Nelder-Mead simplex algorithm
- Classical DoE with (fractional) factorial designs
- Regression trees to handle categorical variables
- Conclusion: Nearly every algorithm can be tuned



Step 2: Integrating Kriging

- Kriging based on matlab's DACE toolbox (Bartz-Beielstein et al., 2004b,c)
- New: space filling designs
- New: sequential approach
- New: coping with noise



SPOT Terminology

- Parameters belong either to (the set of)
  a) Algorithm designs $\mathscr{A}$ or
  b) Problem design $\mathscr{P}$
- Algorithm design: $A \subseteq \mathscr{A}$ with $A = \{\vec{a}_1, \ldots, \vec{a}_n\}$ and $\vec{a}_i = (a_{i1}, \ldots, a_{ik})$, where $n = |A|$ denotes the design size and k the number of parameters
- Two basic tasks. Improve either
  a) Efficiency (tuning) or
  b) Effectivity (robustnes)
- Here: Tuning, i.e., one instance of the problem design $\mathscr{P}$ keep fixed while instances of the algorithm design are varied
- Required: Performance measure $c : \mathscr{A} \times \mathscr{P} \to \mathbb{R}$

Efficiency

- Tuning
- Problems
  - Many factors
  - Real–world problem: complex objective function (simulation) and only small number of function evaluations
  - Theoretical investigations: simple objective function and many function evaluations
- Screening to detect most influential factors

SPO Workflow

1. *Pre-experimental* planning
2. *Scientific* thesis
3. *Statistical* hypothesis
4. Experimental *design*: Problem, constraints, start-/termination criteria, performance measure, algorithm parameters

---

5. *Experiments*
6. Statistical *model* and prediction. Evaluation and visualization
7. Solution good enough?

   Yes: Goto step 8

   No: Improve the design (optimization). Goto step 5

---

8. *Acceptance/rejection* of the statistical hypothesis
9. Objective *interpretation* of the results from the previous step

SPOT Basics

1. Generate initial algorithm design A
2. While termination criterion not true do
   1. Run algorithm on A. This gives n results $\vec{y}$
   2. Sort A w.r.t. $\vec{y}$, so that a_1 is the instance with the best result
   3. Based on A and $\vec{y}$: Build prediction model f for y
   4. Based on f: Predict new values $\vec{y}'$ at unknown design sites A', where $|A'|$ is a large number
   5. Select best (most promising) algorithm designs from A', say A'' with $|A''| << |A'|$
   6. Reevaluate a_1 (improve confidence) and evaluate A'', so that both have the same number of evaluations
   7. $A = A \cup A''$
3. end do

SPOT Region of Interest (ROI)

- *Region of interest* (ROI) files specify the region, over which the algorithm parameters are tuned

```
name low high isint pretty
NPARENTS 1 10 TRUE 'NPARENTS'
NU 1 5 FALSE 'NU'
TAU1 1 3 FALSE 'TAU1'
```

Figure: demo4.roi

SPOT Configuration File

- *Configuration* files (CONF) specify SPOT specific parameters, such as the regression model

```
new=0
defaulttheta=1
loval=1E-3
upval=100
spotrmodel='regpoly2'
spotcmodel='corrgauss'
isotropic=0
repeats=3
...
```

Figure: demo4.m

SPOT Output File

- *Design* files (DES) specify algorithm designs
- Generated by SPOT
- Read by optimization algorithms

```
TAU1 NPARENTS NU TAU0 REPEATS CONFIG SEED STEP
0.210507 4.19275 1.65448 1.81056 3 1 0 1
0.416435 7.61259 2.91134 1.60112 3 2 0 1
0.130897 9.01273 3.62871 2.69631 3 3 0 1
1.65084 2.99562 3.52128 1.67204 3 4 0 1
0.621441 5.18102 2.69873 1.01597 3 5 0 1
1.42469 4.83822 1.72017 2.17814 3 6 0 1
1.87235 6.78741 1.17863 1.90036 3 7 0 1
0.372586 3.08746 3.12703 1.76648 3 8 0 1
2.8292 5.85851 2.29289 2.28194 3 9 0 1
...
```

Figure: demo4.des

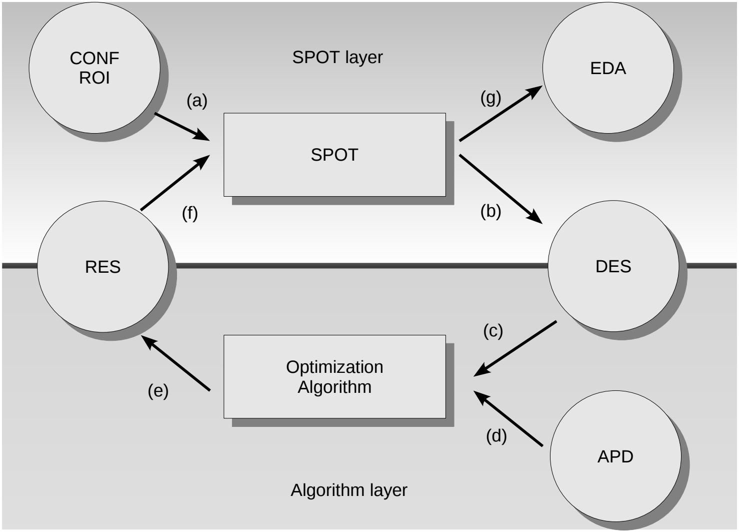
Algorithm: Result File

- Algorithm run with settings from design file
- Algorithm writes *result file* (RES)
- RES files provide basis for many statistical evaluations/visualizations
- RES files read by SPOT to generate prediction model(s)

```
Y NPARENTS FNAME ITER NU TAU0 TAU1 KAPPA NSIGMA RHO DIM CONFIG SEED
3809.15 1 Sphere  500 1.19954 0 1.29436 Inf 1 2 2  1 1
0.00121541  1 Sphere  500 1.19954 0 1.29436 Inf 1 2 2 1 2
842.939 1 Sphere 500 1.19954 0 1.29436 Inf 1 2 2  1 3
2.0174e-005 4 Sphere 500 4.98664 0 1.75367 Inf 1 2 2  2 1
0.000234033 4 Sphere 500 4.98664 0 1.75367 Inf 1 2 2  2 2
1.20205e-007  4 Sphere 500 4.98664 0 1.75367 Inf 1 2 2  2 3
...
```

Figure: demo4.res

SPOT dataflow



Case study: Real-world optimization

- Real-world problem: Prediction
- Data-driven modeling
- New problem, no reference solutions
- How to chose an adequate method?
- How to tune the chosen prediction model?
- Take a look at the problem first
- Here: Prediction of fill levels in stormwater tanks

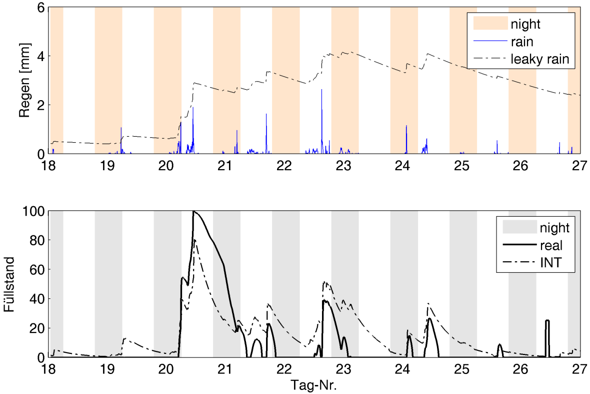
Case Study: Prediction of Fill Levels in Stormwater Tanks

- Based on rain measurements and soil conditions
- Data
  - 150.000 data
  - noisy
  - infeasible

Case Study: Prediction of Fill Levels in Stormwater Tanks

- Goal:
  - Minimize prediction error for 108 days
  - Objective function
  - Fiction of optimization, see Klein (2002)
  - MSE to enable comparisions to results from industry

Case Study: Prediction of Fill Levels



- Problem: Standard and CI-based modeling methods show larger prediction errors when trained on rain data with strong intermittent and bursting behaviour

Case Study: Prediction of Fill Levels

- 6 Methods (many more available):
  1. Neural Networks (NN)
  2. Echo State Networks (ESN)
  3. Nonlinear AutoRegressive models with eXogenous inputs (NARX)
  4. Finite Impulse Response filter (FIR)
  5. Differential equations (ODE)
  6. Integral equations (INT)

Case Study: Prediction of Fill Levels

- Each method has some parameters (here: 2 – 13)
- Problem design vs. algorithm design
- Parameter and factor
  1. Neural Networks (NN): not considered
  2. Echo State Networks (ESN): not considered
  3. Nonlinear AutoRegressive models with eXogenous inputs (NARX): 2, i.e., neurons and delay states
  4. Finite Impulse Response filter (FIR): 5, i.e., evaporation, delay, scaling, decay, length
  5. Differential equations (ODE): 6
  6. Integral equations (INT): 13
- Details: Konen et al. (2009)

Case Study: Prediction of Fill Levels

Table: Factors of the INT-Model. The ODE-Model uses a subset of 6 factors (shaded light gray): $\alpha, \beta, \tau_{\mathrm{rain}}, \Delta, \alpha_L, \beta_L$.

Parameter	Symbol	manuell	Best SPOT	Bereich SPOT
Abklingkonstante Füllstand (Filter g)	α	0.0054	0.00845722	[0,0.02]
Abklingkonstante Filter h	α_H	0.0135	0.309797	{0 ... 1}
Abklingkonstante 'leaky rain'	α_L	0.0015	0.000883692	{0 ... 0.0022}
Einkopplung Regen in Füllstand	β	7.0	6.33486	{0 ... 10}
Einkopplung Regen in 'leaky rain'	β_L	0.375	0.638762	{0 ... 2}
Einkopplung K-Term in Füllstand	h_0	0.5	6.87478	{0 ... 10}
Schwelle für 'leaky rain'	Δ	2.2	7.46989	{0 ... 10}
Flankensteilheit aller Filter	κ	1	1.17136	{0 ... 200}
Zeitverzögerung Füllstand zu Regen	τ_{rain}	12	3.82426	{0 ... 20}
Startzeitpunkt Filter h	τ_{in3}	0	0.618184	{0 ... 5}
Endzeitpunkt Filter h	τ_{out3}	80	54.0925	{0 ... 500}
Endzeitpunkt Filter g	τ_{out}	80	323.975	{0 ... 500}
RMSE		12.723	9.48588	

Case Study: Prediction of Fill Levels in Stormwater Tanks

- SPO in a nutshell
  1. I. Pre-experimental planning
  2. II. Screening
  3. III. Modeling and optimization
- Similar to classical DoE (response surface methods)

Case Study: Prediction of Fill Levels

Step I: Pre-experimental planning

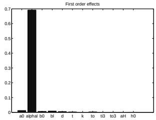
- Test runs, no planning possible
- No optimality conditions applicable
- Detect region of interest intervals, experimental region
- Intervals should courageously be chosen
- Treatment of infeasible factor settings (penalty)

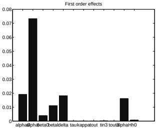
Case Study: Prediction of Fill Levels

Step II: Screening

- Short run time
- Sparse design
- Consider extreme values
- Detect outliers that destroy the SPOT meta-model

- Unbalanced factor effects indicate not correctly specified ROI

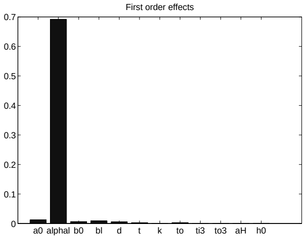




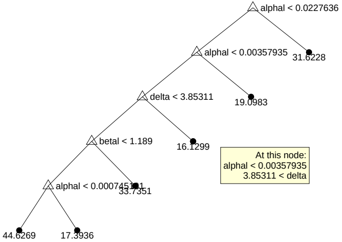
Case Study: Prediction of Fill Levels

Step II: Screening

- Not correctly secified ROIs



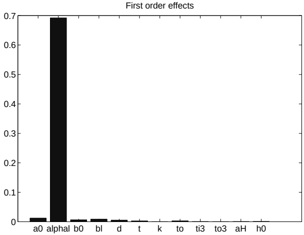
- Regression tree



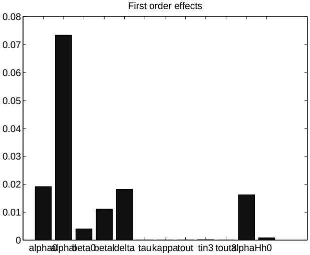
Case Study: Prediction of Fill Levels

Step II: Screening

- Before



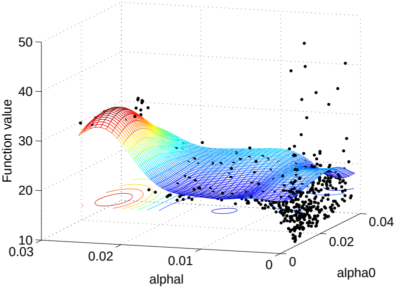
- After



Case Study: Prediction of Fill Levels

Step III: Modeling and Optimization

- Reduced parameter set (INT: from 13 to 6)
- Complex design



Case Study: Prediction of Fill Levels

Result

Table: Comparison. RSME

Method	randomized design	manually chosen	SPOT
FIR	25.42	25.57	20.10
NARX	85.22	75.80	38.15
ODE	39.25	13.60	9.99
INT	31.75	12.72	9.49

Case Study: Prediction of Fill Levels in Stormwater Tanks

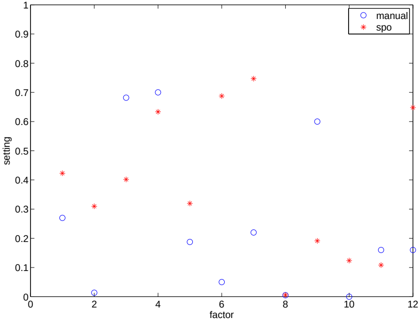
Summary

- Comparison of different prediction methods
- SPOT to determine best parameters for each method
- Problem for standard and CI-based modeling methods: rain data with strong intermittent and bursting behavior
- Models developed specific to the problem show a smaller prediction error
- SPOT applicable to diverse forecasting methods and automates the time-consuming parameter tuning
- Best manual result improved with SPOT by 30%
- SPOT to analyze parameter influence, allows simplification and/or refinement of the model design

Case Study: Prediction of Fill Levels in Stormwater Tanks

Results

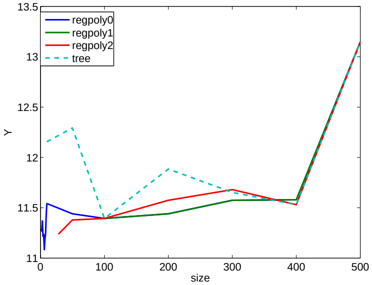
- Ranges
- No bias, no systematic error



Case Study: Prediction of Fill Levels

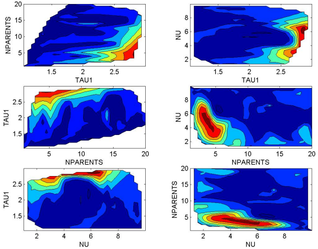
Results

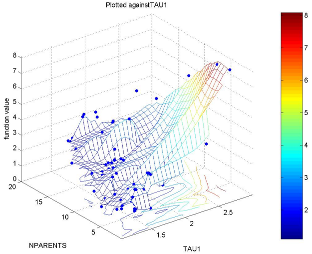
- Design considerations
- How many design points are necessary?
- Initial design size?



SPOT and EDA

- Interaction plots
- Main effect plots
- Regression trees
- Scatter plots
- Box plots
- Trellis plots
- Design plots
- ...





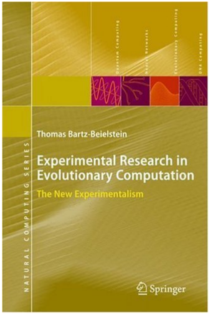
SPOT Open Questions

- Models?
  - (Linear) Regression models
  - Stochastic process models
- Designs?
  - Space filling
  - Factorial
- Enhanced noise handling techniques, e.g., Lasarczyk (2007) implemented OCBA
- Statistical tools
- Significance
- Standards

- SPOT Community:
  - Provide SPOT interfaces for important optimization algorithms
  - Currently: CMAES (Hansen, 1998), PSO, Evolution Strategies
  - Simple and open specification
  - Currently available for several algorithms, more than a dozen applications

- SPO is a methodology — more than just an optimization algorithm (Bartz-Beielstein, 2008)

Updates



- Please check
  `http://www.gm.fh-koeln.de/~bartz`
  for updates, software, etc.
- To appear 2009: Empirical Methods for the Analysis of Optimization Algorithms
- See also Kleijnen's books, e.g., DASE

Bartz-Beielstein, T. (2005). Evolution strategies and threshold selection. In Blesa Aguilera, M. J., Blum, C., Roli, A., and Sampels, M., editors, *Proceedings Second International Workshop Hybrid Metaheuristics (HM'05)*, volume 3636 of *Lecture Notes in Computer Science*, pages 104–115, Berlin, Heidelberg, New York. Springer.

Bartz-Beielstein, T. (2008). How experimental algorithmics can benefit from Mayo's extensions to Neyman-Pearson theory of testing. *Synthese*, 163(3):385–396. DOI 10.1007/s11229-007-9297-z.

Bartz-Beielstein, T., Blum, D., and Branke, J. (2005a). Particle swarm optimization and sequential sampling in noisy environments. In Hartl, R. and Doerner, K., editors, *Proceedings 6th Metaheuristics International Conference (MIC2005)*, pages 89–94, Vienna, Austria.

Bartz-Beielstein, T., Chmielewski, A., Janas, M., Naujoks, B., and Scheffermann, R. (2006). Optimizing door assignment in ltl-terminals by evolutionary multiobjective algorithms. In Fogel, D. B. e. a., editor, *Proc. 2006 Congress on Evolutionary Computation (CEC'06) within Fourth IEEE World Congress on Computational Intelligence (WCCI'06), Vancouver BC*, pages 348–354, Piscataway NJ. IEEE Press.

Bartz-Beielstein, T., de Vegt, M., Parsopoulos, K. E., and Vrahatis, M. N. (2004a). Designing particle swarm optimization with regression trees. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–173/04, Universität Dortmund, Germany.

Bartz-Beielstein, T., Lasarczyk, C., and Preuß, M. (2005b). Sequential parameter optimization. In McKay, B. et al., editors, *Proceedings 2005 Congress on Evolutionary Computation (CEC'05), Edinburgh, Scotland*, volume 1, pages 773–780, Piscataway NJ. IEEE Press.

Bartz-Beielstein, T. and Naujoks, B. (2004). Tuning multicriteria evolutionary algorithms for airfoil design optimization. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–159/04, Universität Dortmund, Germany.

Bartz-Beielstein, T., Parsopoulos, K. E., and Vrahatis, M. N. (2004b). Analysis of particle swarm optimization using computational statistics. In Simos, T.-E. and Tsitouras, C., editors, *Proceedings International Conference Numerical Analysis and Applied Mathematics (ICNAAM)*, pages 34–37, Weinheim, Germany. Wiley-VCH.

Bartz-Beielstein, T., Parsopoulos, K. E., and Vrahatis, M. N. (2004c). Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis and Computational Mathematics (ANACM)*, 1(2):413–433.

Bartz-Beielstein, T., Preuß, M., and Markon, S. (2005c). Validation and optimization of an elevator simulation model with modern search heuristics. In Ibaraki, T., Nonobe, K., and Yagiura, M., editors, *Metaheuristics: Progress as Real Problem Solvers*, Operations Research/Computer Science Interfaces, pages 109–128. Springer, Berlin, Heidelberg, New York.

de Vegt, M. (2005). Einfluss verschiedener Parametrisierungen auf die Dynamik des Partikel-Schwarm-Verfahrens: Eine empirische Analyse. Interner Bericht der Systems Analysis Research Group SYS–3/05, Universität Dortmund, Fachbereich Informatik, Germany.

Hansen, N. (1998). *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie*. Mensch & Buch, Berlin, Germany.

Kleijnen, J. P. C. (1987). *Statistical Tools for Simulation Practitioners*. Marcel Dekker, New York NY.

Klein, G. (2002). The fiction of optimization. In Gigerenzer, G. and Selten, R., editors, *Bounded Rationality: The Adaptive Toolbox*, pages 103–121. MIT Press, Cambridge MA.

Konen, W., Zimmer, T., and Bartz-Beielstein, T. (2009). Optimierte modellierung von füllständen in regenüberlaufbecken mittels ci-basierter parameterselektion. *at – Automatisierungstechnik*. Im Druck.

Lasarczyk, C. W. G. (2007). *Genetische Programmierung einer algorithmischen Chemie*. PhD thesis, Technische Universitï¿½t Dortmund.

Markon, S., Kita, H., Kise, H., and Bartz-Beielstein, T., editors (2006). *Modern Supervisory and Optimal Control with Applications in the Control of Passenger Traffic Systems in Buildings*. Springer, Berlin, Heidelberg, New York.

Mehnen, J., Michelitsch, T., Bartz-Beielstein, T., and Henkenjohann, N. (2004). Systematic analyses of multi-objective evolutionary algorithms applied to real-world problems using statistical design of experiments. In Teti, R., editor, *Proceedings Fourth International Seminar Intelligent Computation in Manufacturing Engineering (CIRP ICME'04)*, volume 4, pages 171–178, Naples, Italy. CIRP ICME'04.

Mehnen, J., Michelitsch, T., Lasarczyk, C. W. G., and Bartz-Beielstein, T. (2005). Multiobjective evolutionary design of mold temperature control using DACE for parameter optimization. In Pfützner, H. and Leiss, E., editors, *Proceedings Twelfth International Symposium Interdisciplinary Electromagnetics, Mechanics, and Biomedical Problems (ISEM 2005)*, volume L11-1, pages 464–465, Vienna, Austria. Vienna Magnetics Group Reports.

Stoean, C., Preuss, M., Gorunescu, R., and Dumitrescu, D. (2005). Elitist Generational Genetic Chromodynamics - a New Radii-Based Evolutionary Algorithm for Multimodal Optimization. In McKay, B. et al., editors, *Proc. 2005 Congress on Evolutionary Computation (CEC'05)*, volume 2, pages 1839 – 1846, Piscataway NJ. IEEE Press.

Tosic, M. (2006). Evolutionäre Kreuzungsminimierung. Diploma thesis, University of Dortmund, Germany.

Weinert, K., Mehnen, J., Michelitsch, T., Schmitt, K., and Bartz-Beielstein, T. (2004). A multiobjective approach to optimize temperature control systems of moulding tools. *Production Engineering Research and Development, Annals of the German Academic Society for Production Engineering*, XI(1):77–80.