UNIVERSITÄT PADERBORN

AUTOMATIC CONTROL GROUP

PARTICLE SWARM OPTIMIZATION ASSISTED BY SURROGATES

A HYBRID APPROACH TO GLOBAL OPTIMIZATION

Marvin Meyer & Hendrik Vater

Overview

- Fundamentals
  - ➢ Global Optimization
  - ➢ Particle Swarm Optimization
  - ➢ Bayesian Optimization
- Methods & Evaluation
  - ➢ Hybrid Approach PSO & Bayesian Optimization
  - ➢ Data Buffer
- Conclusion & Outlook

Global Optimization

- Find parameters/inputs that optimize a criterion/function in a constrained search space
- Occurs in all kinds of disciplines, universal applications
  - ➢ Completing a task with the least amount of resources
  - ➢ Finding some optimal model
- Trivial for simple examples, but complexity can grow dramatically due to:
  - ➢ Increasing number of parameters
  - ➢ Non-convex functions / local optima
- Exploration vs. exploitation (in contrast to local optimization)

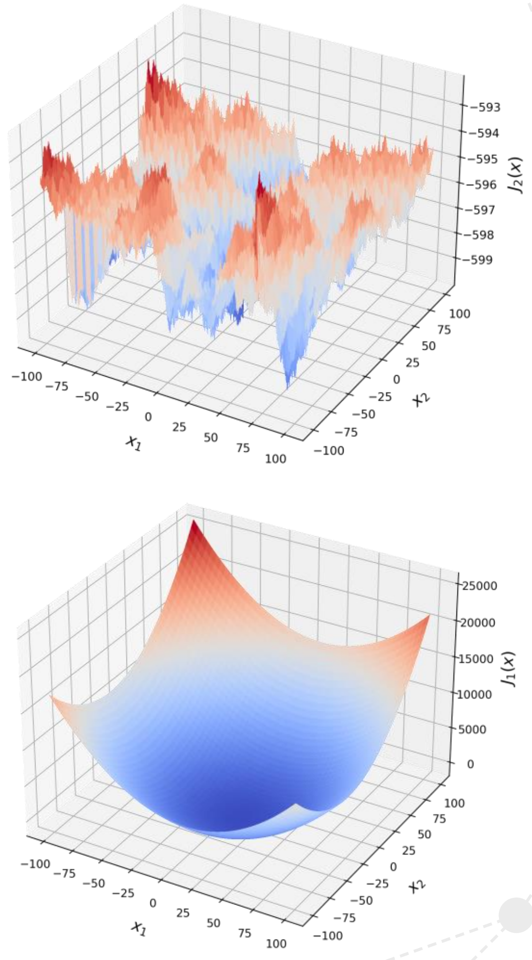


Fig: Comparison of two functions

Particle Swarm Optimization (PSO, see [2])

- Mimics swarm intelligence to solve optimization problems
- Each particle moves through search space according to:

$$\vec{V}_{n+1} = \underbrace{\omega \cdot \vec{V}_n}_{\text{Old velocity}} + \underbrace{c_1 \cdot \vec{U}_1 \otimes \left(\vec{P}_n - \vec{X}_n\right)}_{\text{Personal best}} + \underbrace{c_2 \cdot \vec{U}_2 \otimes \left(\vec{L}_n - \vec{X}_n\right)}_{\text{Swarm best}}$$

$$\vec{X}_{n+1} = \vec{X}_n + \vec{V}_{n+1}$$

- Different topologies change flow of information
  - ➢ Global, ring, adaptive random, etc.

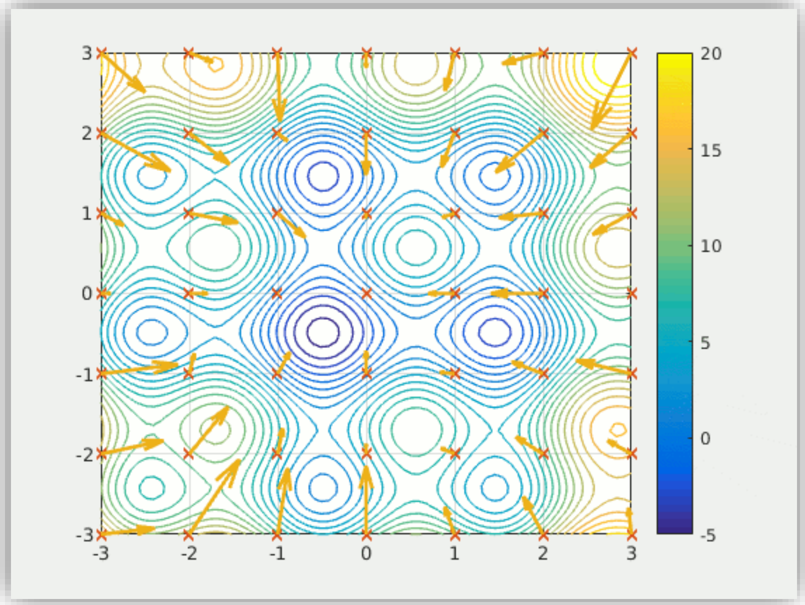


Fig: Example of a particle swarm searching for the global optimum [5]

Standard PSO 2011 (SPSO2011, see [2])

- Proposed improvement to PSO
- Sample uniformly from a hypersphere

$$\vec{C}_n = \frac{\vec{X}_n + \overrightarrow{P'}_n + \overrightarrow{L'}_n}{3}$$

$$\vec{V}_{n+1} = \omega \cdot \vec{V}_n + \mathcal{H}\left(\vec{C}_n, \left\|\vec{C}_n - \vec{X}_n\right\|\right) - \vec{X}_n$$

$$\vec{X}_{n+1} = \vec{X}_n + \vec{V}_{n+1}$$

- Does not use any old data other than the best points

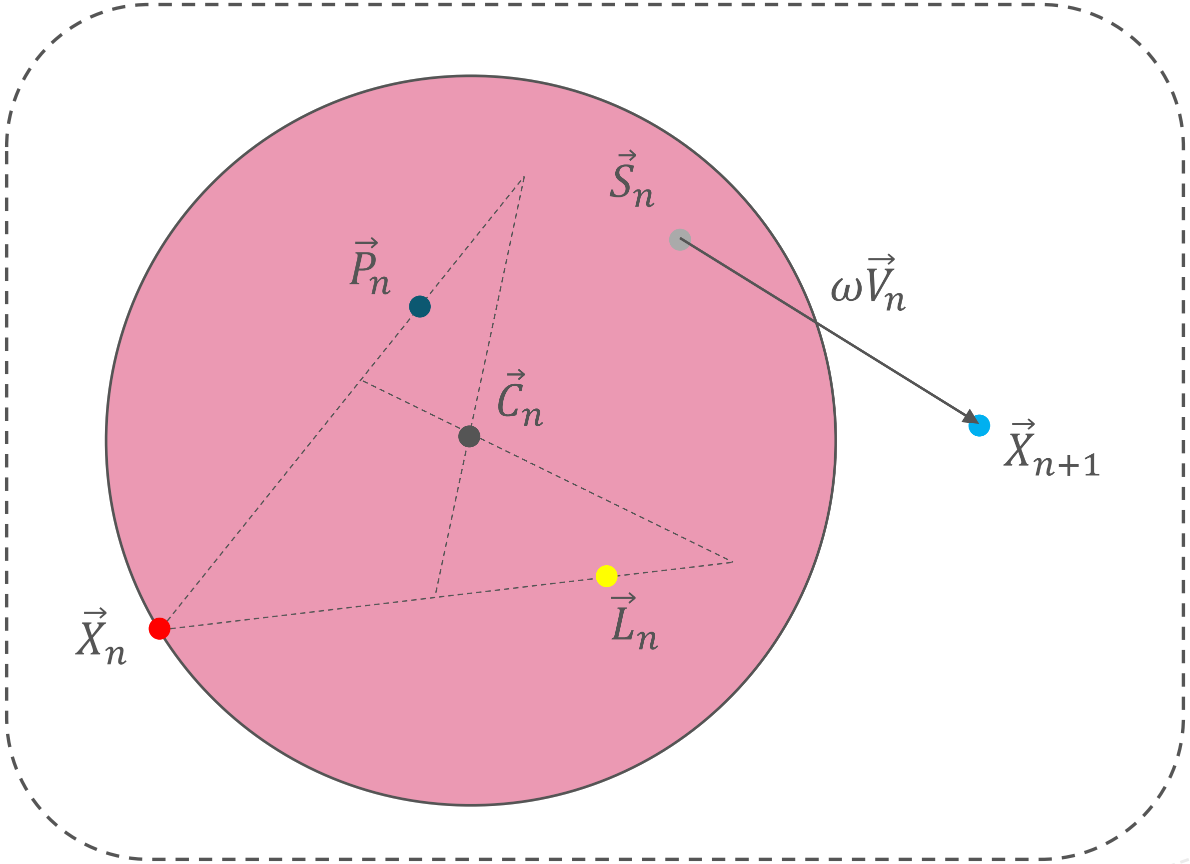


Fig: Illustration of the SPSO2011 update rule [2]

Bayesian Optimization (for maximization)

- Model as an approximation of the real function
- Focus on gaussian processes
  - ➢ Model consists of mean and uncertainty/variance
- Based on the model an acquisition function is created
  - ➢ Computationally faster to evaluate
  - ➢ Yields a point of potential improvement
  - ➢ High uncertainty and high mean (for maximization) constitute to a good value for the acquisition function
- Alternating between fit and acquisition function optimization

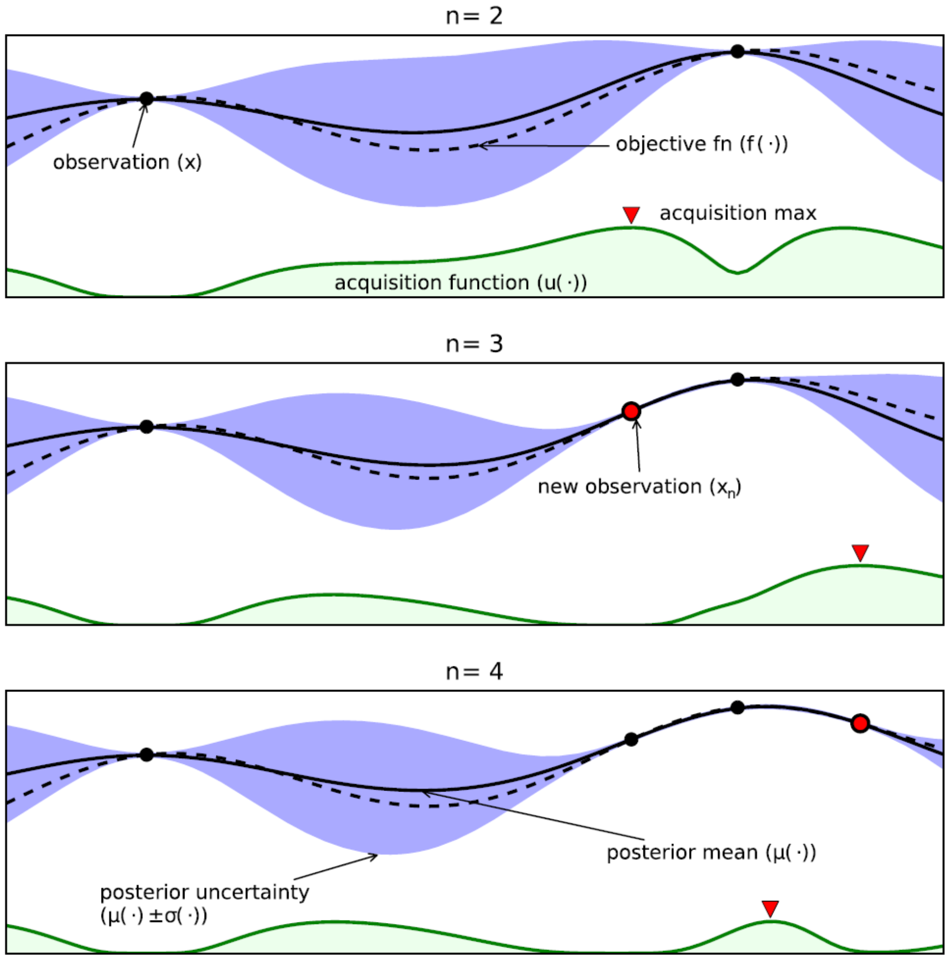
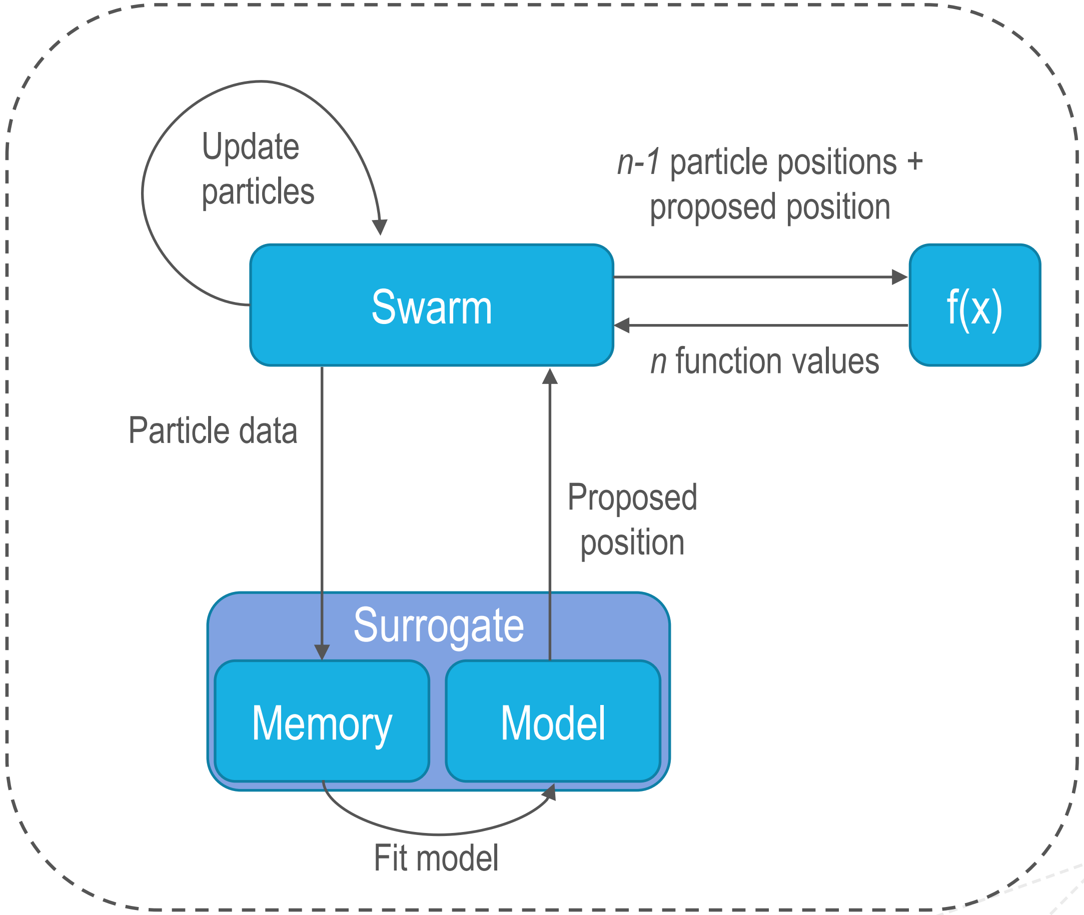


Fig: Example for a maximization problem [3]

SPSO2011 + Bayesian Optimization

- Keep more of the gathered points in memory
- Swarm gathers data
- Surrogate is fit using this data
- Worst particle of the swarm replaced with result of the acquisition function optimization
- Replacement can also be repeated for the m worst points



Another Take on the Topic

- Paper using the PSO with a gaussian process-based surrogate by Jakubik et al [4]
- 3 variants which were compared to SPSO2011 and Bayesian optimization

- Benchmark test CEC2013:
  - Standardized test, including 28 minimization problems
  - Search space range: $[-100, 100]^D$
  - Function evaluations: $100 \cdot D \quad (1000$ for $D = 10)$
  - 51 runs per function
- Jakubik's Variant A3 outperforms all their other versions

Results for SPSO2011 and Variants

Mean over 51 runs

Func	Opt	SPSO	SPSO + 1 Bayesian	Jakubik SPSO	Jakubik A3
1	-1400	-651,44	-946,36	-7,89	**-1395,01**
2	-1300	1,5E+07	1,5E+07	1,4E+08	**1,2E+07**
3	-1200	2,8E+09	2,8E+09	1,1E+11	**2,3E+09**
4	-1100	**3,3E+04**	3,7E+04	4,0E+05	4,0E+05
5	-1000	-567,30	**-626,03**	-308,40	-381,12
6	-900	-814,04	-827,45	-764,74	**-848,43**
7	-800	-711,90	**-715,51**	-632,23	-684,63
8	-700	-679,26	-679,29	**-679,29**	-679,19
9	-600	-590,14	-590,24	-589,45	**-594,45**
10	-500	-385,22	**-476,34**	-264,25	-451,97
11	-400	-327,45	-325,58	-309,30	**-355,27**
12	-300	-225,65	-226,34	-215,63	**-247,02**
13	-200	-127,80	-127,96	-110,52	**-146,96**
14	-100	2011,62	2040,22	1957,85	**1086,95**

Func	Opt	SPSO	SPSO + 1 Bayesian	Jakubik SPSO	Jakubik A3
15	0	2136,09	2162,76	2220,26	**1615,87**
16	100	202,52	202,52	202,65	**202,46**
17	200	415,35	380,63	416,11	**301,89**
18	300	507,99	479,54	521,40	**401,50**
19	400	513,44	513,57	628,37	**507,32**
20	500	604,25	**604,21**	604,31	604,27
21	600	1162,25	**1150,31**	1176,77	1473,88
22	700	3141,37	3143,18	3090,08	**2247,24**
23	800	3212,69	3227,08	3368,55	**2646,50**
24	900	1228,09	1226,85	1231,84	**1210,11**
25	1000	**1326,01**	1326,11	1330,75	1334,72
26	1100	1415,51	1399,33	**1398,62**	1409,85
27	1200	1896,74	1904,12	1922,83	**1641,31**
28	1300	2218,45	2253,96	2346,69	**2043,26**

Data buffer to reduce the amount of data

- Problems with global surrogate:
  - Good fit in high dimensions needs lots of data
  - Fit itself takes a lot of time with increasing number of data points
  - Areas where the swarm does not explore stay somewhat uncertain

- Use a data buffer to reduce the amount of data for the fit

- Type 1: Time Buffer
  - “First in, first out”
  - Data in the surrogate memory follows the swarm

- Type 2: Value Buffer
  - “Best in, worst out”
  - New values are compared with the worst elements of the buffer

Results for SPSO2011 with Buffer

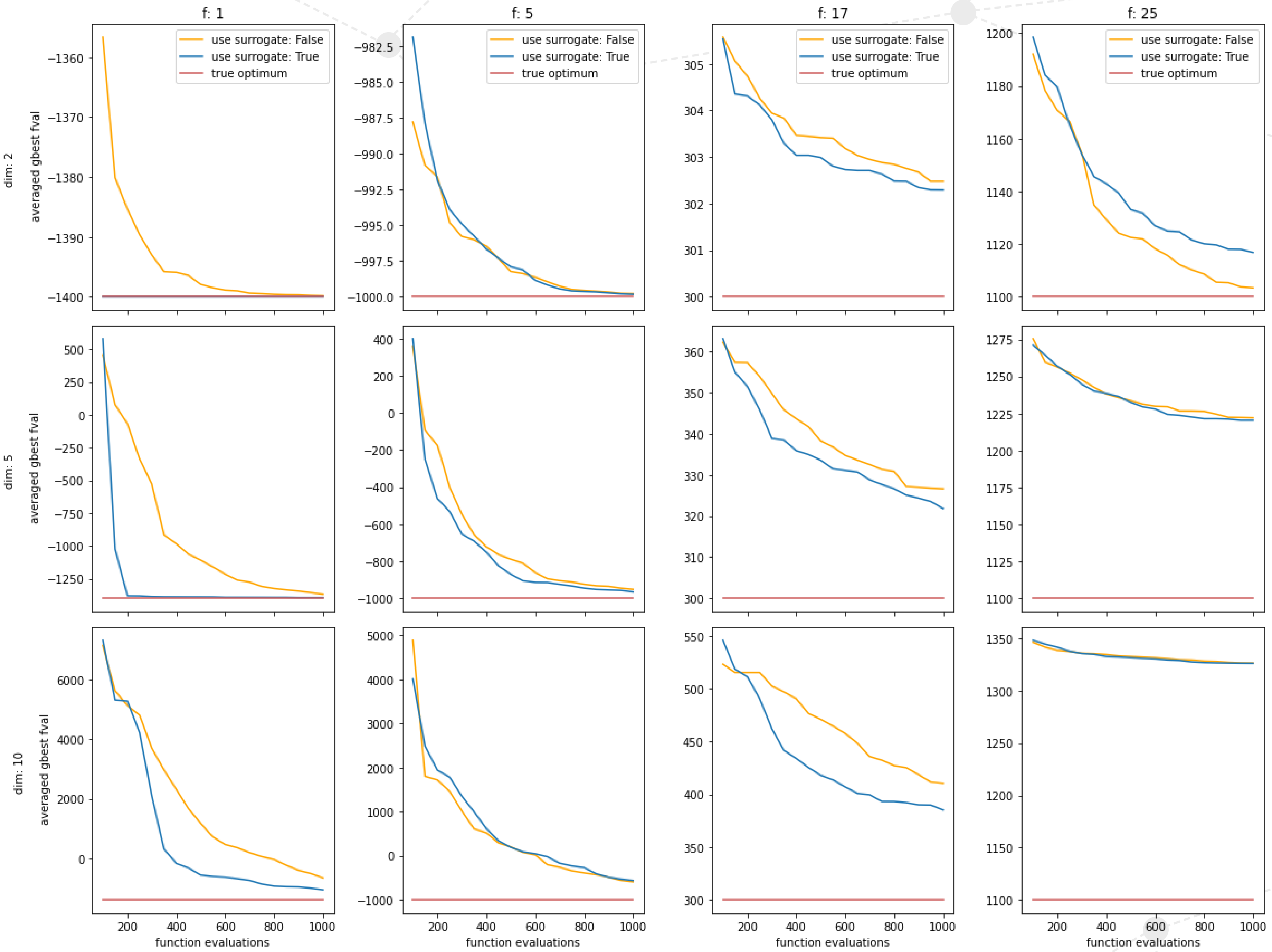
Mean over 51 runs

Func	Opt	SPSO	SPSO + 5 Bayesian + Time Buffer	SPSO + 5 Bayesian + Value Buffer	Jakubik A3
1	-1400	-651,44	-1272,60	-1255,33	**-1395,01**
2	-1300	1,5E+07	1,6E+07	1,5E+07	**1,2E+07**
3	-1200	2,8E+09	2,7E+09	2,7E+09	**2,3E+09**
4	-1100	3,3E+04	3,6E+04	**3,1E+04**	4,0E+05
5	-1000	-567,30	-586,46	**-676,45**	-381,12
6	-900	-814,04	-843,75	**-886,80**	-848,43
7	-800	-711,90	-711,78	**-719,40**	-684,63
8	-700	-679,26	**-679,27**	-679,26	-679,19
9	-600	-590,14	-590,08	-590,03	**-594,45**
10	-500	-385,22	-489,17	**-489,36**	-451,97
11	-400	-327,45	-341,16	-348,96	**-355,27**
12	-300	-225,65	-232,34	**-249,45**	-247,02
13	-200	-127,80	-128,77	-146,57	**-146,96**
14	-100	2011,62	2019,93	2014,32	**1086,95**

Func	Opt	SPSO	SPSO + 5 Bayesian + Time Buffer	SPSO + 5 Bayesian + Value Buffer	Jakubik A3
15	0	2136,09	2192,53	2180,21	**1615,87**
16	100	202,52	202,46	**202,42**	202,46
17	200	415,35	372,77	373,78	**301,89**
18	300	507,99	471,47	477,80	**401,50**
19	400	513,44	512,64	511,10	**507,32**
20	500	604,25	604,23	**604,21**	604,27
21	600	1162,25	1124,19	**1114,98**	1473,88
22	700	3141,37	3124,40	3135,47	**2247,24**
23	800	3212,69	3227,38	3227,00	**2646,50**
24	900	1228,09	1227,09	1227,05	**1210,11**
25	1000	1326,01	1328,74	**1325,47**	1334,72
26	1100	1415,51	1396,74	**1393,98**	1409,85
27	1200	1896,74	1913,74	1920,73	**1641,31**
28	1300	2218,45	2240,62	2182,84	**2043,26**

Convergence Behavior

- Time-based Buffer
- One Bayesian optimization step
- Averaged over 15 runs



Conclusion

- Github repository with evaluation framework (https://github.com/upb-lea/PSOAS)
- Baseline SPSO2011 implementation
- Surrogate addition fairly effective for predictable / convex examples
- No significant improvement over the baseline for other tested functions
  - ➢ A large portion of the surrogate propositions are not helpful currently
- Only applicable if the function evaluation is expensive

Outlook

- Looking at Jakubik there should be room for improvement
- Global surrogate approach:
  - Filtering the data is necessary
  - Recognizing appropriate points needs to be improved
- Local surrogate approach:
  - Constraints, regularization are necessary
  - How to choose these constraints?

Sources

[1] https://mathworld.wolfram.com/GlobalOptimization.html (Accessed 06.10.2021 13:54)

[2] M. Zambrano-Bigiarini, M. Clerc and R. Rojas, "Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements." (2013)

[3] B. Shahriari, et al. "Taking the human out of the loop: A review of Bayesian optimization.“ (2015)

[4] J. Jakubik, A. Binding, and S. Feuerriegel. "Directed particle swarm optimization with Gaussian-process-based function forecasting.“ (2021)

[5] https://commons.wikimedia.org/wiki/File:ParticleSwarmArrowsAnimation.gif (Accessed 10.10.2021 18:57)