WOST 2023
Benchmark of One-Click-Optimizer (OCO)

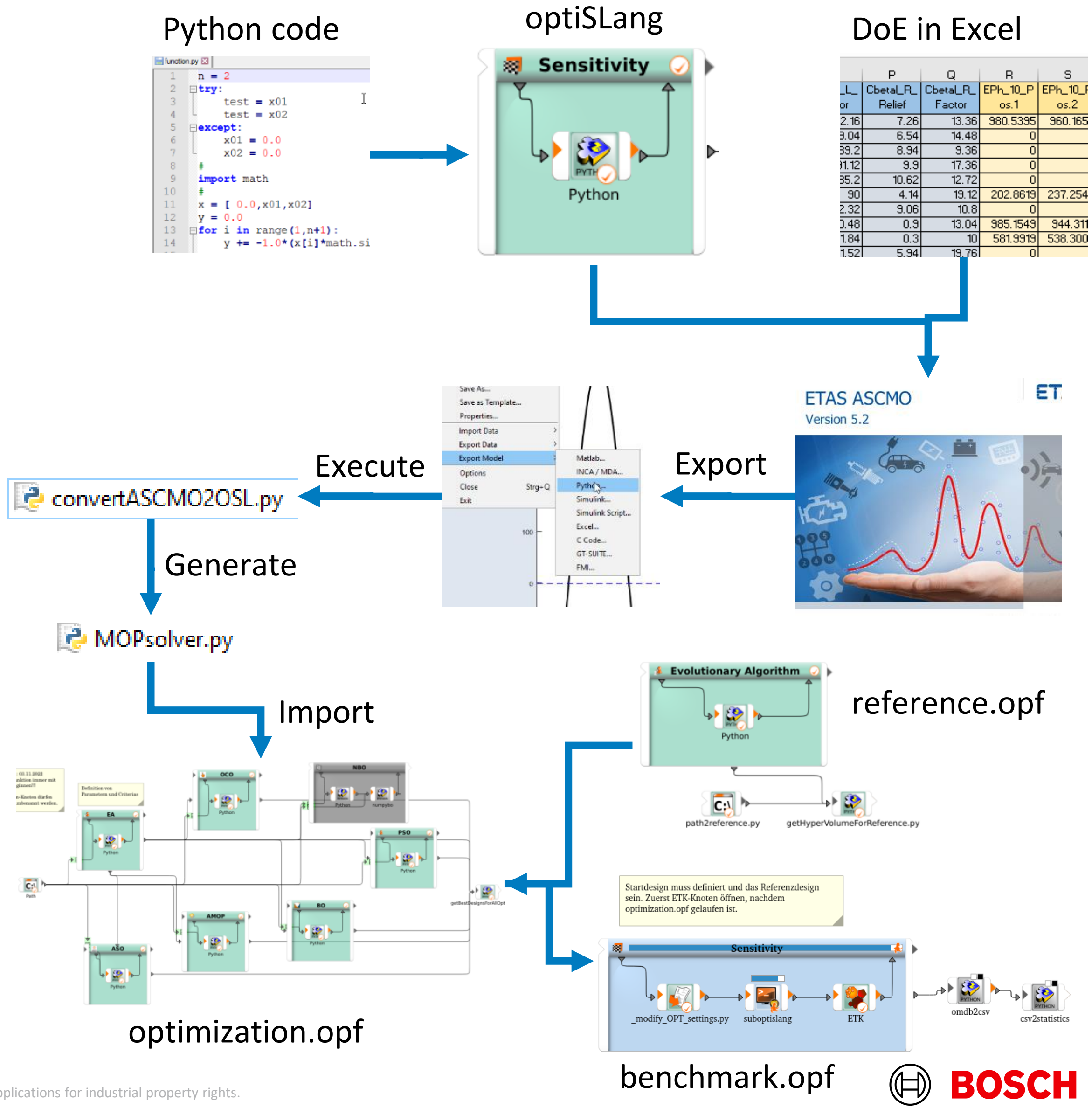
Schirrmacher, Bosch CR/AME3, 22.06.2023

BOSCH

Benchmark of One-Click-Optimizer
Agenda

Introduction
General Approach

- As simulation model an ASCMO (ETAS) model is used in order to be independent from optiSLang and Stochos machine learning models.
- The data for ASCMO can be generated by a Python code plus a sensitivity study or a Design of Experiment in Excel format.
- The trained Gaussian Process models from ASCMO are exported as Python and Matlab code.
- The Python/Matlab code is modified and the simulation model “MOPsolver.py/.m” is created.
- The “MOPsolver.py/.m” is used as solver for different algorithms in optimization.opf. The reference optimum is found in reference.opf.
- A sensitivity study with slightly modified parameters of the algorithms is performed in order to get a statistical evaluation of the different algorithms.



Benchmark of One-Click-Optimizer
Agenda

Selection of Optimization Algorithms

Algorithms inside optiSLang and from external sources

- Nature Inspired Optimization Algorithms
  - Evolutionary Algorithm (EA)
  - Particle Swarm Optimization Algorithms (PSO)
- Adaptive Optimization Algorithms
  - Adaptive Single-Objective Optimization Algorithm (ASO)
  - Adaptive Multi-Objective Optimization Algorithm (AMO)
  - Adaptive Metamodel of Optimal Prognosis (AMOP)
  - Bayesian Optimization (BO)
- Hybrid Optimization Algorithms
  - One Click Optimization Algorithm (OCO)

- SIGOPT (SIGOPT)
  - Mixture from global and Bayesian optimization from the company Intel
- Black Box Optimization from Bosch (BCAI)
  - Space Filling by Sobol-Sequences
  - MBORE: Multi-objective Bayesian Optimization by Density-Ratio Estimation
- CR optimizer from Bosch (CROPT)
  - NSGA II algorithm
  - Special features, not suitable for that benchmark

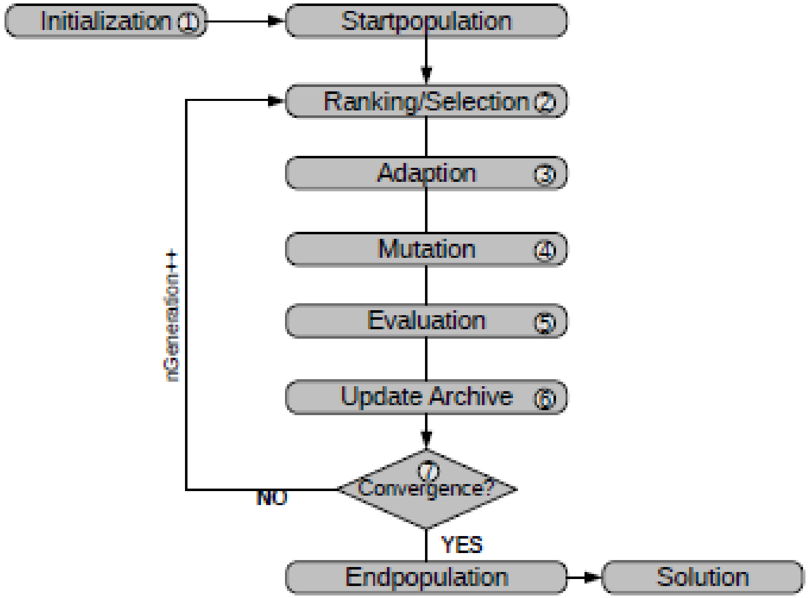
Selection of Optimization Algorithms

Nature Inspired Optimization Algorithms

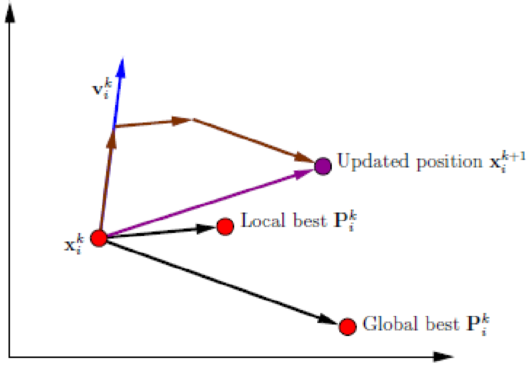
Nature Inspired Optimization

- The Evolutionary Algorithm is well known from the mode of operation for many user.
- The EA was/is mostly used as default optimization algorithm at Bosch. The global search is selected as default.
- The Particle Swarm Optimization is more difficult to understand like the “swarm behaviour”.
- The other NIO algorithms like Stochastic Design Improvement and Covariance Matrix Adaption are not used.

Evolutionary Algorithm



Particle Swarm Optimization



*) Dynardo/2023R1-Help/ optiSLang_methods.pdf

Selection of Optimization Algorithms
Adaptive Optimization Methods (direct solver calls)

- The **Adaptive Single-Objective Method** is a gradient-based method that employs advanced refinement methods to provide the global optima. It requires a minimum number of design points to build the Kriging metamodel. Failed design points are treated as inequality constraints. The Adaptive Single-Objective method is available for input parameters that are continuous.
- **Procedure**
  - Create Kriging metamodel based on an optimal space filling Latin Hypercube sampling
  - Apply MISQP on the metamodel
  - Successive refinement steps of DoE+ Kriging, to optimize again
- The **Adaptive Multiple-Objective Method** method is an iterative algorithm that allows you to either generate a new sample set or use an existing set, providing a more refined approach than the Screening method. It uses the same general approach as MOGA, but applies the Kriging error predictor to reduce the number of samples needed to find the global optimum. The Adaptive Multiple-Objective method is available only for continuous input parameters.

*) Dynardo/2023R1/
optiSLang_Users_Guide.pdf

Selection of Optimization Algorithms

Adaptive Optimization Methods

Adaptive Metamodel of Optimal Prognosis

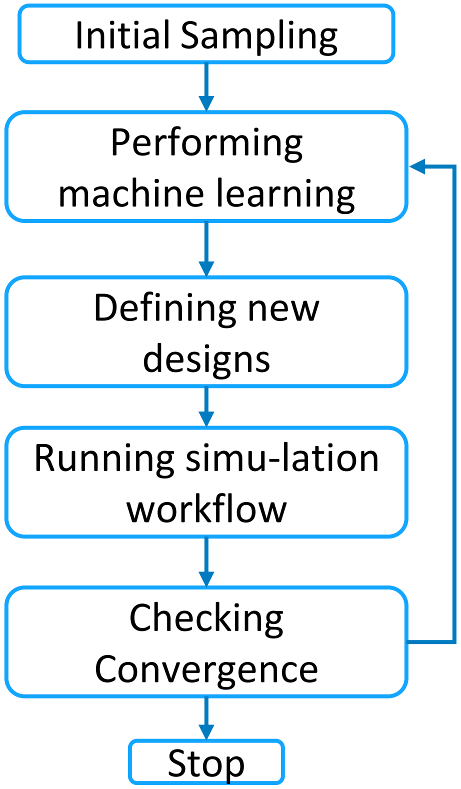
Probabilistic Inference for Bayesian Optimization (PI-BO)

AMOP

PI-BO

- The **A**daptive **M**etamodel of **O**ptimal **P**rognosis is an iterative meta-modelling approach. The **AMOP** can also be used for optimi-zation, if two settings are selected.
  - Refinement Type=Local
  - Importance of optimization criteria=100%
- The default settings for MOP were used for **AMOP**.

- The **PI-BO** is developed by the company Probaligence and is integrated in optiSLang.
- **PI-BO** is based on the **Bayesian optimization** method and uses the probabilistic properties of the DIM-GP metamodel to select new designs based on the greatest potential for design improvements while taking model uncertainty into account.
- **PI-BO** is especially suitable for engineering problems where the evaluation of the designs goes along with high computational costs.



*) Dynardo/2023R1/ optiSLang_Users_Guide.pdf

BOSCH

Selection of Optimization Algorithms
Hybrid Optimization Methods

- The **One Click Optimizer (OCO)** provides an efficient hybrid optimization strategy that comes with only one major setting to be tuned, the maximum number of design evaluations. Depending on the type and number of input parameters and the defined optimization criteria, the optimizer automatically selects the most suitable optimization algorithms with their most appropriate settings to solve the optimization problem. The ability to dynamically switch between optimization algorithms and to run multiple algorithms simultaneously makes OCO one of the most reliable and efficient optimization strategies. OCO is a surrogate assisted optimization strategy, using capabilities of the Metamodel of Optimal Prognosis (MOP) for function approximation to significantly speed up the optimization process.
- The default settings for MOP were used for OCO.

*) Dynardo/2023R1/
optiSLang_Users_Guide.pdf

Selection of Optimization Algorithms
Definition of maximum number of designs

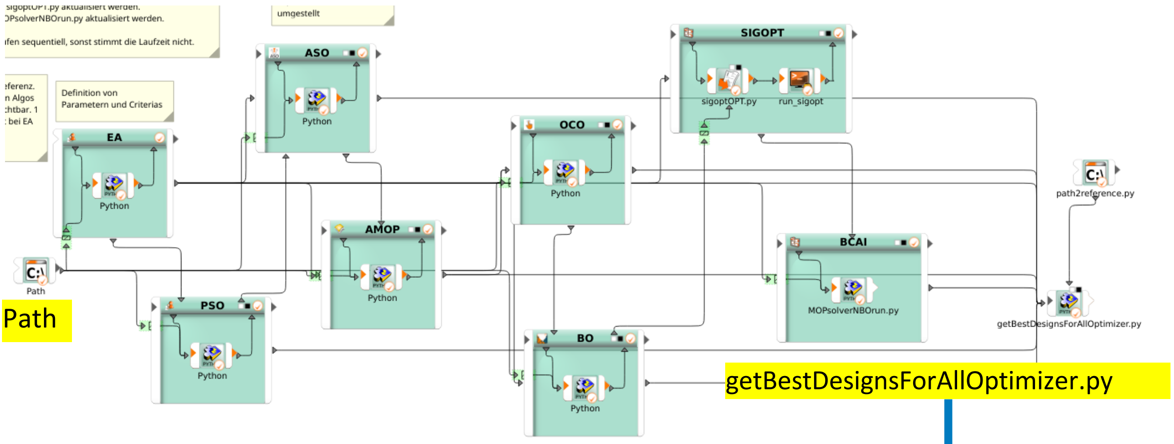
- The four main parameters of the optimization algorithms are
  - A: Maximum number of designs for EA, PSO, ASO, AMO
  - B: Maximum number of designs for BO
  - C: Maximum number of designs for AMOP
  - D: Maximum number of designs for OCO

- The naming of the header lines of the subsequent slides of the examples is
  - setting_A_B_C_D

Benchmark of One-Click-Optimizer
Agenda

Workflow for Single-Objective-Optimization
optiSLang project: optimization.opf

- The Path node contains the path to the file "MOPsolver.py" which represents the simulation.
- The following algorithms are used for a fixed number of designs.
  - EA	Evolutionary Algorithm
  - PSO	Particle Swarm Optimizer
  - ASO	Adaptive Single-objective Optimizer
  - AMOP	Adaptive Metamodel of Optimal Prognosis
  - PI-BO	Bayesian Optimizer
  - OCO	One Click Optimizer
  - SIGOPT	Optimizer from Intel
  - BCAI	Optimizer from former BCAI (Bosch)
- The Python code getBestDesignsForAllOptimizer.py extracts important results for the comparison.



getBestDesignsForAllOptimizer.erg

```
EA_OptimumValue  = -8.37257318e+02
EA_OptimumDiff   =  3.68985271e-02
EA_MaxCstValue   =  2.11410673e+01
EA_DesignDiff    =  7.23862266e-01
EA_NumDesigns    =  2.00000000e+02
EA_CompTime      =  3.26750112e+00

PSO_OptimumValue = -8.34281934e+02
PSO_OptimumDiff  =  3.01228277e+00
PSO_MaxCstValue  =  2.11598703e+01
```

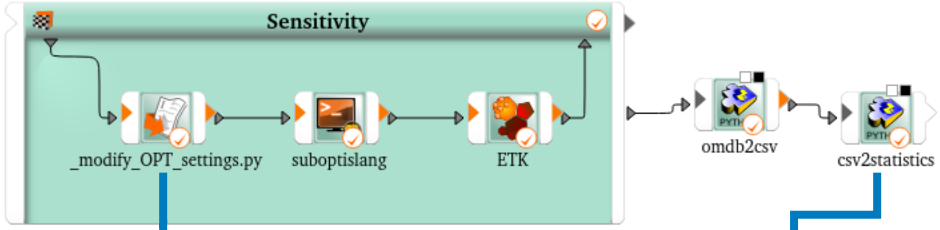

CR/AME3 | 2023-06-22

BOSCH

Workflow for Single-Objective-Optimization

optiSLang project: benchmark.opf

- The Python code in _modify_OPT_settings.py changes the settings of some algorithms.
- When starting the optislang.opf by suboptislang, this Python code modifies the settings of the optimizer.
- After having run up to 100 different optislang.opf projects, a statistics about mean, dev, min and max values is exported as well as a ranking.
- The ranking evaluates only the difference to the optimum value. The mean value, the minimum and the maximum value of all runs are taken as criteria. The lower the value of the ranking, the better is the optimizer.




```
# Globale Parameter
numMaxParallel   = 1
maxNumDesigns    = 200
#
# NIO-EA
EA_search         = 'BALANCED'
EA_fitness        = 'WEIGHTED_SUM'
EA_cstHandling    = 'PENALTY_RANK'
EA_mutationRate = 0.0
#
# NIO-PSO
PSO_search          = 'BALANCED'
PSO_fitness         = 'WEIGHTED_SUM'
PSO_cstHandling     = 'PENALTY_RANK'
PSO_weight_begin = 0.0
PSO_weight_end    = 0.0
PSO_person_begin = 0.0
PSO_person_end    = 0.0
PSO_global_begin = 0.0
PSO_global_end    = 0.0
#
# AMOP
#
# BO
#
# OCO
#
# ASO
ASO_num_max_cycles      = 0.0
ASO_num_start_ml        = 0.0
ASO_num_max_reduct      = 0.0
ASO_retained_domain     = 0.0
ASO_domain_reduction    = 0.0
```

```
NumDesigns_AMOP_mean     =   1.67440000E+02
NumDesigns_BO_mean       =   6.00000000E+01
NumDesigns_OCO_mean      =   1.12000000E+02
NumDesigns_ASO_mean      =   1.73240000E+02

#
CompTime_EA_mean         =   3.25288250E+00
CompTime_PSO_mean        =   2.35205256E+00
CompTime_AMOP_mean       =   4.62021411E+01
CompTime_BO_mean         =   2.99895847E+03
CompTime_OCO_mean        =   1.41020167E+01
CompTime_ASO_mean        =   1.12578569E+01

#
------------------------------------------
OptimumValue_EA_dev      =   9.10231349E+01
OptimumValue_PSO_dev     =   8.15292490E+01
OptimumValue_AMOP_dev    =   3.58435762E+01
```

```
Ranking:
EA     = 29
PSO    = 28
AMOP   = 34
BO     = 49
OCO    = 33
ASO    = 19
```


BOSCH

Benchmark of One-Click-Optimizer
Agenda

Example for Single-Objective-Optimization

RC18: Pressure Vessel Design

https://www.sciencedirect.com/science/article/pii/S2210650219308946

- The pressure vessel design is an official benchmark example for real-world applications.
- The optimization is a mixed-integer problem with 4 design variables and 4 constraints.
- There is a mismatch in the optimum value. The theoretical value should be 5885.3, but the best value is
  - f = 6059.714
  - x1 = 13 , x2 = 7, x3 = 42.09844, x4 = 176.63659

True Global Optimality of the Pressure Vessel Design Problem: A Benchmark for Bio-Inspired Optimisation Algorithms

Xin-She Yang, Christian Huyck, Mehmet Karamanoglu, Nawaz Khan
School of Science and Technology, Middlesex University,
The Burroughs, London NW4 4BT, UK.

Swarm and Evolutionary Computation 56 (2020) 100693

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

ELSEVIER

A test-suite of non-convex constrained optimization problems from the real-world and some baseline results

Abhishek Kumar [a], Guohua Wu [b], Mostafa Z. Ali [c], Rammohan Mallipeddi [d], Ponnuthurai Nagaratnam Suganthan [e,*], Swagatam Das [f]

[a] Department of Electrical Engineering, Indian Institute of Technology (BHU), Varanasi, Varanasi, 221005, India
[b] School of Traffic and Transportation Engineering, Central South University, Changsha, 410075, China
[c] School of Computer Information Systems, Jordan University of Science & Technology, 22110, Jordan
[d] School of Electronics Engineering, Kyungpook National University, Daegu, 41566, Republic of Korea
[e] School of Electrical Electronic Engineering, Nanyang Technological University, Singapore, 639798
[f] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India

2.3.4. Pressure vessel design [43].

The main objective of this problem is to optimize the welding cost, material, and forming of a vessel. This problem contains four constraints which are needed to be satisfied, and four variables are used to calculate the objective function: shell thickness (z_1), head thickness (z_2), inner radius (x_3), and length of the vessel without including the head (x_4). This problem can be stated as.

Minimize:

$$f(\bar{x}) = 1.7781 z_2 x_3^2 + 0.6224 z_1 x_3 x_4 + 3.1661 z_1^2 x_4 + 19.84 z_1^2 x_3 \quad (24)$$

subject to:

$$g_1(\bar{x}) = 0.00954 x_3 \le z_2,$$

$$g_2(\bar{x}) = 0.0193 x_3 \le z_1,$$

$$g_3(\bar{x}) = x_4 \le 240,$$

$$g_4(\bar{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 \le -1296000.$$

where:

$$z_1 = 0.0625 x_1,$$

$$z_2 = 0.0625 x_2.$$

with bounds:

$$10 \le x_4, x_3 \le 200$$

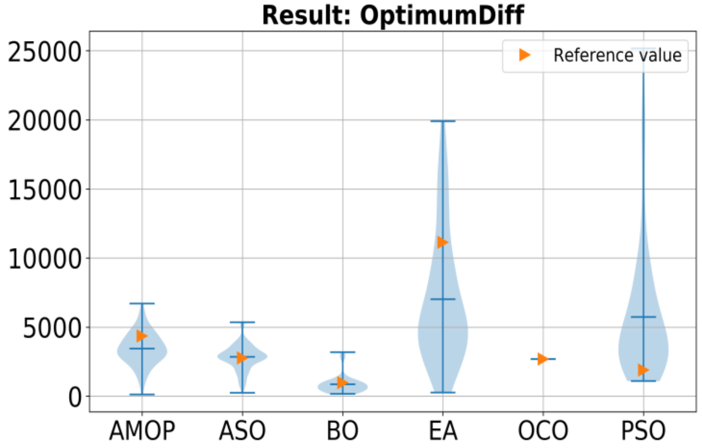
$$1 \le x_2, x_1 \le 99 \quad \text{(integer variables).}$$

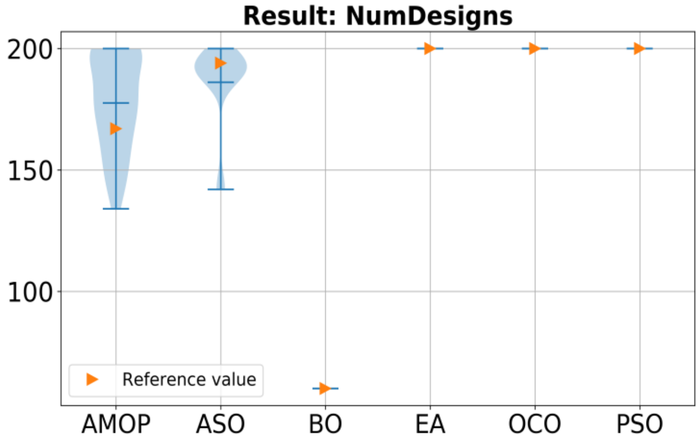
BOSCH

Example for Single-Objective-Optimization

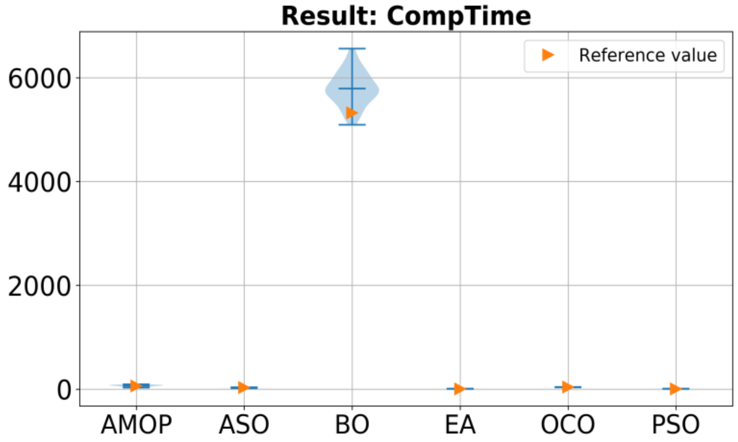
RC18: Results – setting_200_60_200_200

- The best optimizer is BO which finds an optimum value which is quite close to the best value.
- The OCO has a fixed seed of the initial sampling, which leads to identical results. This bug will be fixed in optiSLang 2023R2.
- EA and PSO have a wide range of optimal solutions.
- The BO requires a long computation time.





Ranking	
Algorithm	200/60/200/200
BO	8
ASO	12
AMOP	12
OCO	13
EA	18
PSO	19

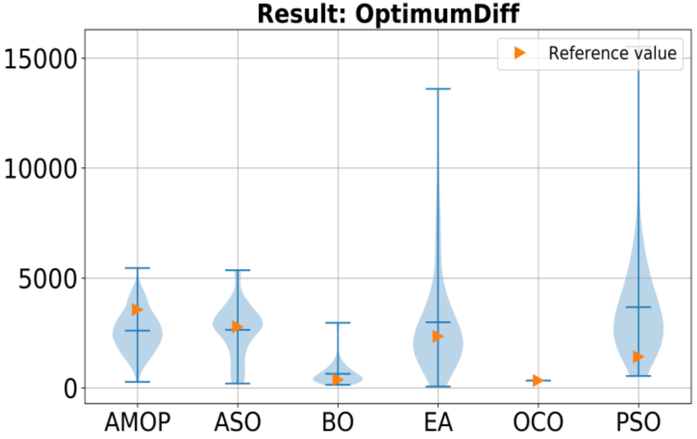


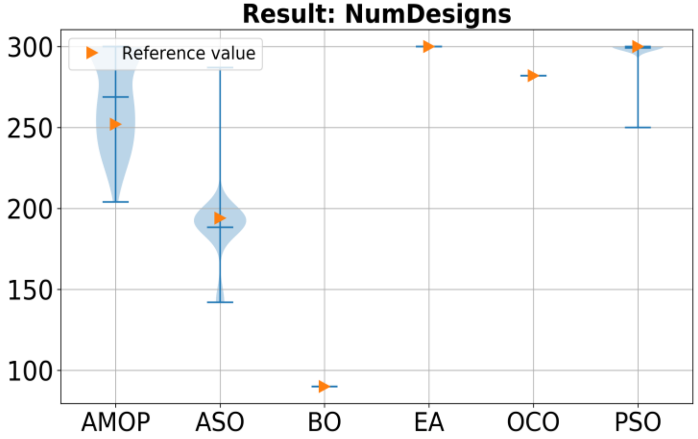
BOSCH

Example for Single-Objective-Optimization

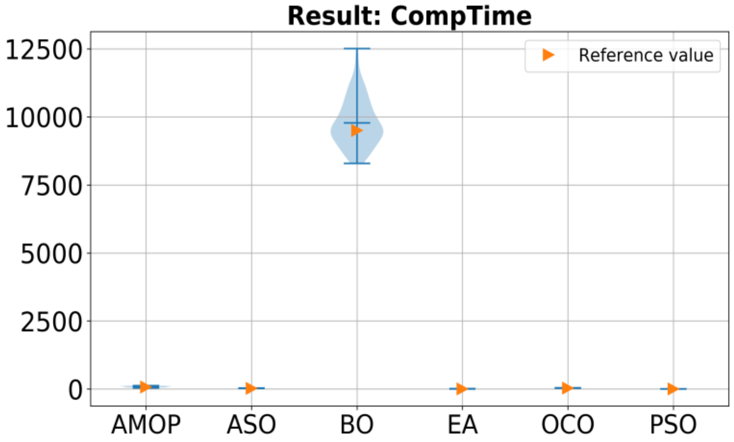
RC18: Results – setting_300_90_300_400

- The best optimizer is again BO.
- The OCO performs much better with the double number of designs (200 → 400).
- All other algorithms do not find the optimum well.





Ranking	
Algorithm	300/90/300/400
BO	9
OCO	10
ASO	13
EA	14
AMOP	14
PSO	21



Example for Single-Objective-Optimization

RC15: Speed reducer

- The speed reducer is an official benchmark example for real-world applications.
- The optimization is a problem with 7 continuous design variables and 11 constraints.
- The optimum is
  - f = 2999.17063
  - x7 = 5.28636135
  - x6 = 3.34996302
  - x5 = 7.73038815
  - x4 = 7.30199647
  - x3 = 17.0183333
  - x2 = 0.700166667
  - x1 = 3.50246144

https://www.sciencedirect.com/science/article/pii/S2210650219308946

Swarm and Evolutionary Computation 56 (2020) 100693

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

A test-suite of non-convex constrained optimization problems from the real-world and some baseline results

Abhishek Kumar[a], Guohua Wu[b], Mostafa Z. Ali[c], Rammohan Mallipeddi[d], Ponnuthurai Nagaratnam Suganthan[e,*], Swagatam Das[f]

[a] Department of Electrical Engineering, Indian Institute of Technology (BHU), Varanasi, Varanasi, 221005, India
[b] School of Traffic and Transportation Engineering, Central South University, Changsha, 410075, China
[c] School of Computer Information Systems, Jordan University of Science & Technology, 22110, Jordan
[d] School of Electronics Engineering, Kyungpook National University, Daegu, 41566, Republic of Korea
[e] School of Electrical Electronic Engineering, Nanyang Technological University, Singapore, 639798
[f] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India

Minimize:

$$f(\bar{x}) = 0.7854x_2^2x_1(14.9334x_3 - 43.0934 + 3.3333x_3^2)$$
$$+0.7854(x_5x_7^2 + x_4x_6^2) - 1.508x_1(x_7^2 + x_6^2) + 7.477(x_7^3 + x_6^3)$$

subject to:

$$g_1(\bar{x}) = -x_1x_2^2x_3 + 27 \leq 0,$$
$$g_2(\bar{x}) = -x_1x_2^2x_3^2 + 397.5 \leq 0,$$
$$g_3(\bar{x}) = -x_2x_6^4x_3x_4^{-3} + 1.93 \leq 0,$$
$$g_4(\bar{x}) = -x_2x_7^4x_3x_5^{-3} + 1.93 \leq 0,$$
$$g_5(\bar{x}) = 10x_6^{-3}\sqrt{16.91 \times 10^6 + (745x_4x_2^{-1}x_3^{-1})^2} - 1100 \leq 0,$$
$$g_6(\bar{x}) = 10x_7^{-3}\sqrt{157.5 \times 10^6 + (745x_5x_2^{-1}x_3^{-1})^2} - 850 \leq 0,$$
$$g_7(\bar{x}) = x_2x_3 - 40 \leq 0,$$
$$g_8(\bar{x}) = -x_1x_2^{-1} + 5 \leq 0,$$
$$g_9(\bar{x}) = x_1x_2^{-1} - 12 \leq 0,$$
$$g_{10}(\bar{x}) = 1.5x_6 - x_4 + 1.9 \leq 0,$$
$$g_{11}(\bar{x}) = 1.1x_7 - x_5 + 1.9 \leq 0,$$

with bounds:

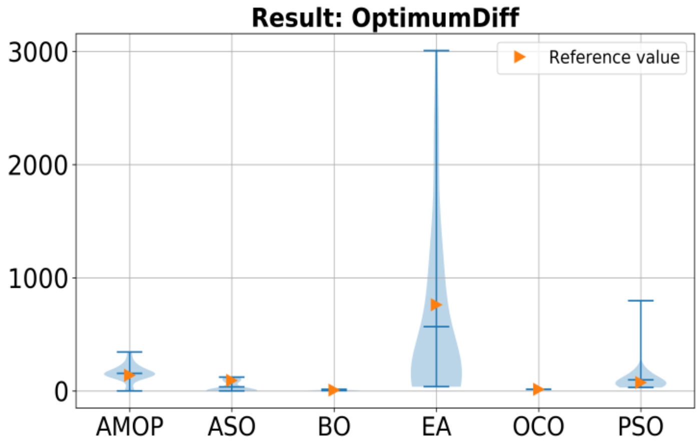
$$0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 2.6 \leq x_1 \leq 3.6,$$
$$5 \leq x_7 \leq 5.5, 7.3 \leq x_5, x_4 \leq 8.3, 2.9 \leq x_6 \leq 3.9.$$

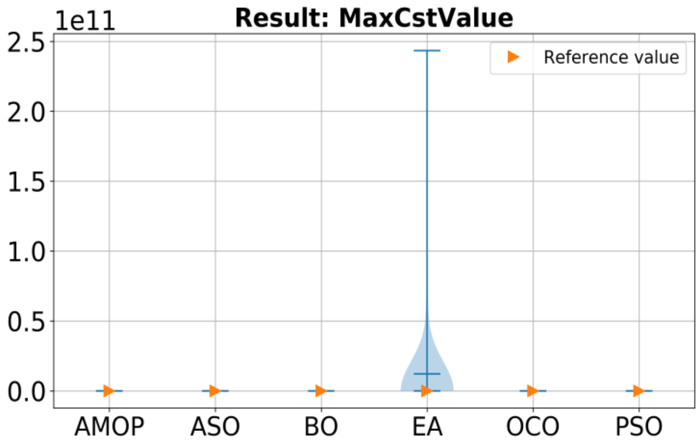
BOSCH

Example for Single-Objective-Optimization

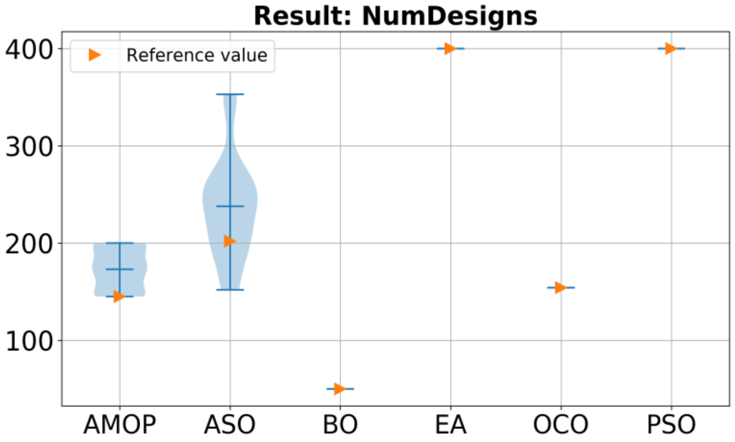
RC15: Results – setting_400_50_200_600

- BO and OCO show the best results followed by ASO and AMOP.
- PSO and EA often do not find a valid optimum design which partly leads to very high violation of the constraints – especially for the EA.
- AMOP, ASO and OCO stopped be-fore reaching the maximum number of designs.
- BO requires a very high computation time – about 3 hours compared to several minutes of other algorithms.





Ranking	
Algorithm	400/50/200/600
BO	6
OCO	7
ASO	8
AMOP	10
PSO	14
EA	18



BOSCH

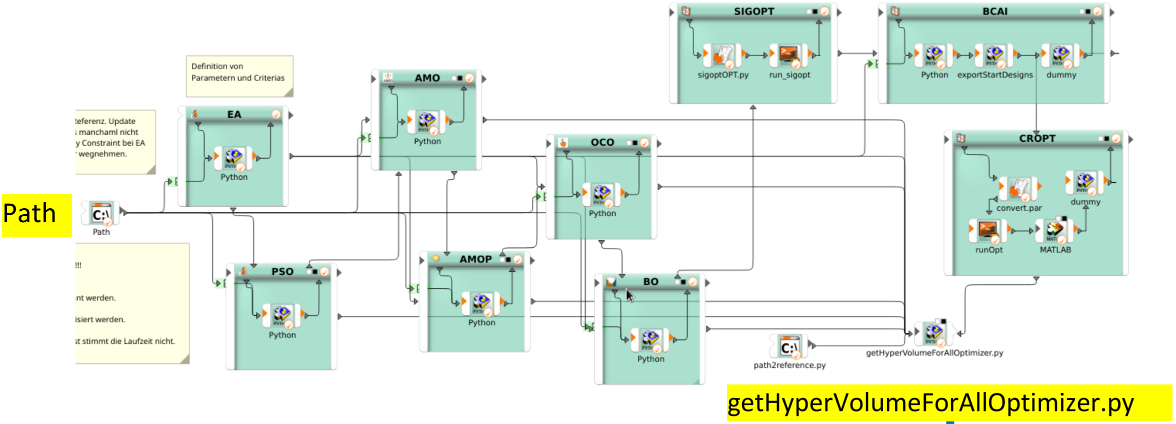
Benchmark of One-Click-Optimizer
Agenda

BOSCH

Workflow for Multi-Objective-Optimization

optiSLang project: optimization.opf

- The Path node contains the path to the file "MOPsolver.py" which represents the simulation.
- The following algorithms are used
  - EA	Evolutionary Algorithm
  - PSO	Particle Swarm Optimizer
  - AMO	Adaptive Multi-objective Optimizer
  - AMOP	Adaptive Metamodel of Optimal Prognosis
  - BO	Bayesian Optimizer
  - OCO	One Click Optimizer
  - SIGOPT	Optimizer from Intel
  - BCAI	Optimizer from former BCAI (Bosch)
  - CROPT	NSGA-II-Optimizer from CR (Bosch)
- The Python code getHyperVolumeForAllOptimizer.py extracts important results for the comparison



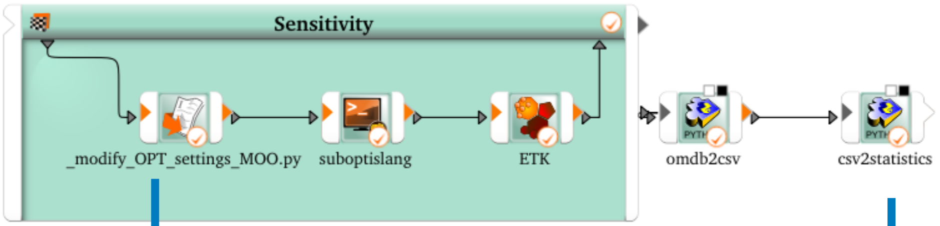

```
EA_Hypervolume        =  1.50169085e+03
EA_DiffHypervol       = -1.78232155e+01
EA_NumDesigns         =  1.77000000e+02
EA_numVioCst          =  0.00000000e+00
EA_CompTime           =  7.55598998e+00

PSO_Hypervolume       =  1.49880263e+03
PSO_DiffHypervol      = -2.07114385e+01
PSO_NumDesigns        =  1.24000000e+02
PSO_numVioCst         =  0.00000000e+00
```


BOSCH

Workflow for Multi-Objective-Optimization

optiSLang project: benchmark.opf

- The Python code in _modify_OPT_settings_MOO.py changes settings of some algorithms.
- When starting the optislang.opf by suboptislang, this Python code modifies the settings of the optimizer.
- After having run up to 100 different optislang.opf projects, a statistics about mean, dev, min and max values is exported as well as a ranking.
- The ranking includes only the difference to the optimal hypervolume. The mean value, the minimum and the maximum value of all runs are taken as criteria. The lower the value of the ranking, the better is the optimizer.




```
# Globale Parameter
numMaxParallel   = 1
maxNumDesigns    = 400
#
# NIO-EA
EA_search        = 'BALANCED'
EA_fitness       = 'PARETO_DOMINANCE'
EA_cstHandling   = 'PENALTY_RANK'
EA_mutationRate  = 0.0
#
# NIO-PSO
PSO_search        = 'BALANCED'
PSO_fitness       = 'PARETO_DOMINANCE'
PSO_cstHandling   = 'PENALTY_RANK'
PSO_weight_begin  = 0.0
PSO_weight_end    = 0.0
PSO_person_begin  = 0.0
PSO_person_end    = 0.0
PSO_global_begin  = 0.0
PSO_global_end    = 0.0
#
# AMOP
#
# BO
#
# OCO
#
# AMO_NODE (Anhang wegen Doppeldeutigl
AMO_max_pareto_percentage  = 0.0
AMO_conv_stab_percentage   = 0.0
```

```
#
CompTime_EA_ref            =  2.51905820E+01
CompTime_PSO_ref           =  2.41305146E+01
CompTime_AMOP_ref          =  7.31366346E+02
CompTime_OCO_ref           =  1.80854240E+02
CompTime_AMO_ref           =  1.80854240E+02

#
Hypervolume_EA_mean        =  1.50572577E+03
Hypervolume_PSO_mean       =  1.49735560E+03
Hypervolume_AMOP_mean      =  1.51637025E+03
Hypervolume_OCO_mean       =  1.51224463E+03
Hypervolume_AMO_mean       =  1.47847778E+03

#
DiffHypervol_EA_mean       =  1.37883021E+01
DiffHypervol_PSO_mean      =  2.21584748E+01
```

```
Ranking:
EA     = 29
PSO    = 28
AMOP   = 34
BO     = 49
OCO    = 33
```

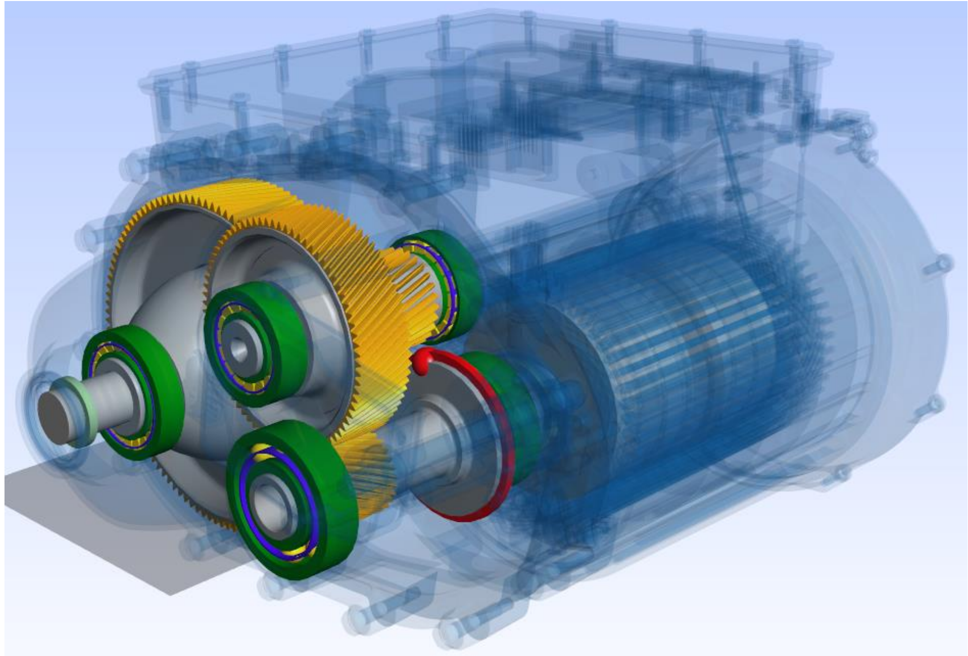
Benchmark of One-Click-Optimizer
Agenda

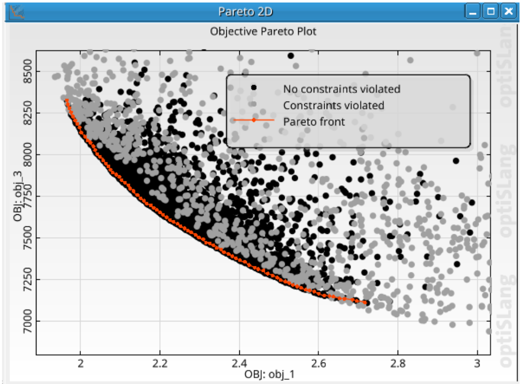
BOSCH

Examples for Multi-Objective-Optimization

Acoustic Properties in Gear Simulation

- The NVH behaviour of gear systems also depends on the micro geometry.
- 16 design variables of the micro geometry were used to calculate 32 response variables based on different loading conditions.
- The 32 response variables were used to define
  - 2 objective functions to minimize
  - 4 constraints



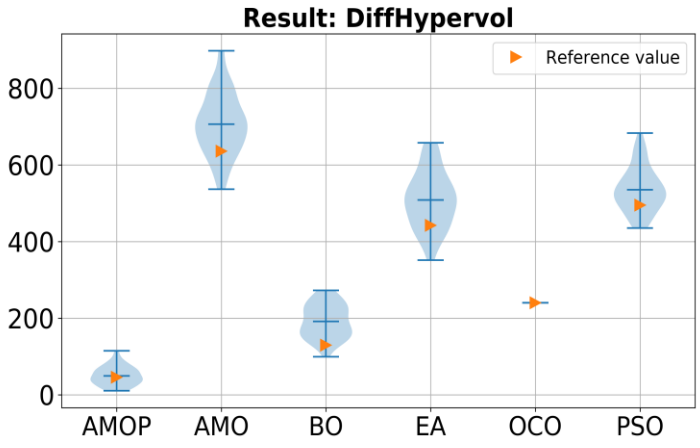


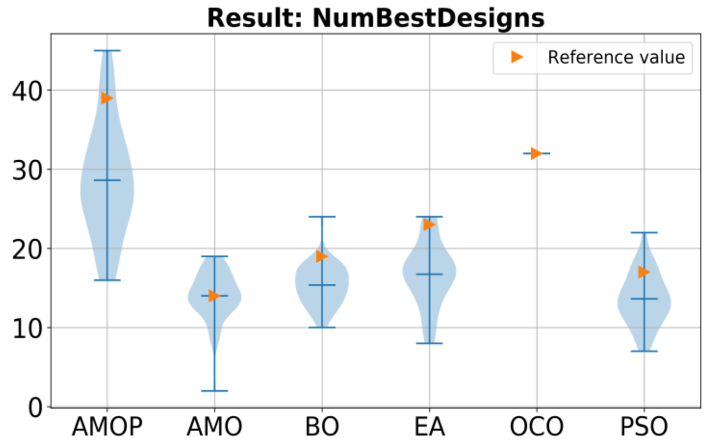
$$Hypervolume: v = 1134.6$$
$$Hypervolumereferenz: rf1 = 3$$
$$rf2 = 8500$$

Examples for Multi-Objective-Optimization

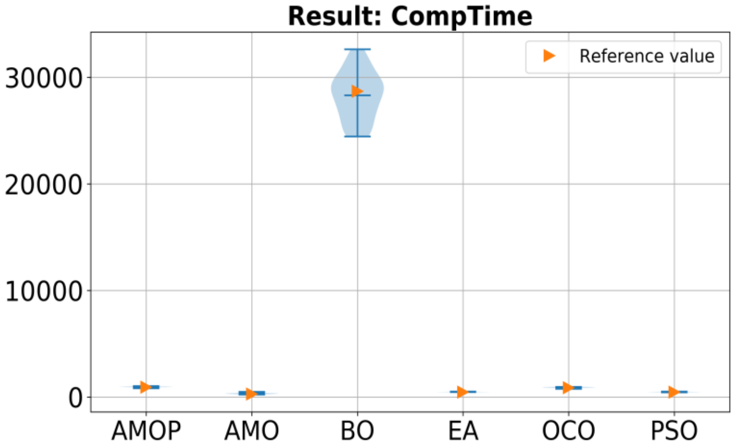
Gear Simulation: Results – setting 400_80_200_400

- Best algorithm is AMOP and which has a small difference to the reference hypervolume and which has a large number of Pareto designs.
- BO and OCO are following in the ranking.
- AMO, EA and PSO do not show a good solution. The variance of the solution is quite high.





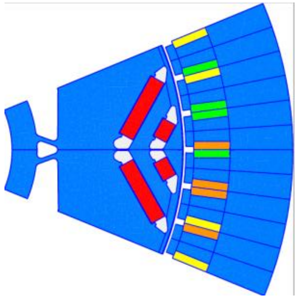
Ranking	
Algorithm	400/80/200/400
AMOP	6
BO	10
OCO	11
EA	15
PSO	19
AMO	24



BOSCH

Examples for Multi-Objective-Optimization

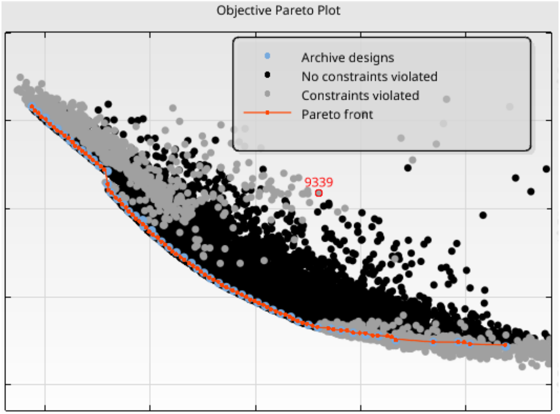
Performance of an eMachine



- The geometry of the eMachine has 27 parameters. Two parameters have discrete values
  - p : number of pole pairs
  - q: slots per pole per phase
- No geometry check was taken into account.
- The original optimization problem consists of two objective functions and 2 constraints.
- Because the calculation of the hypervolume does not allow negative values for the objective function which comes from the maximization of the maximum power, an offset of 220 was selected and a minimization of the difference to 220.
- 30 optimization runs were performed.

Criteria

Name	Type	Expression	Criterion	Limit	
obj_mat_Cost	Objective	mat_Cost	MIN		125.599
obj_P_max	Objective	220-P_max	MIN		95.6283
constr_M_max	Constraint	M_max-200	≥	0	140.508
constr_I_AKS	Constraint	400-I_AKS	≥	0	176.87



$$Hypervolume: v = 12807.0$$

$$Hypervolumereferenz: mat\_Cost = 120$$

$$P\_max = 220$$

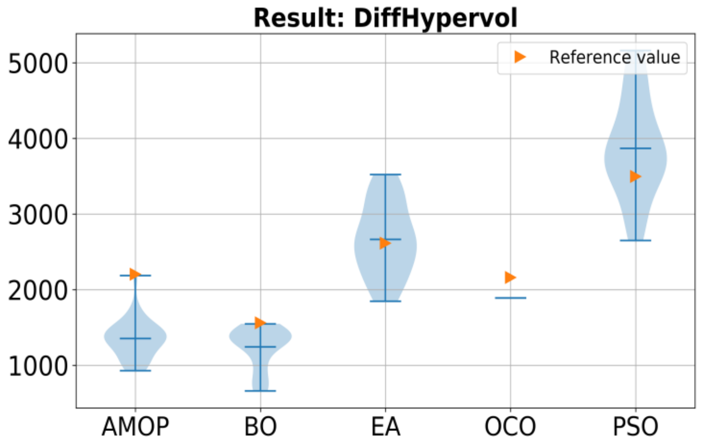
BOSCH

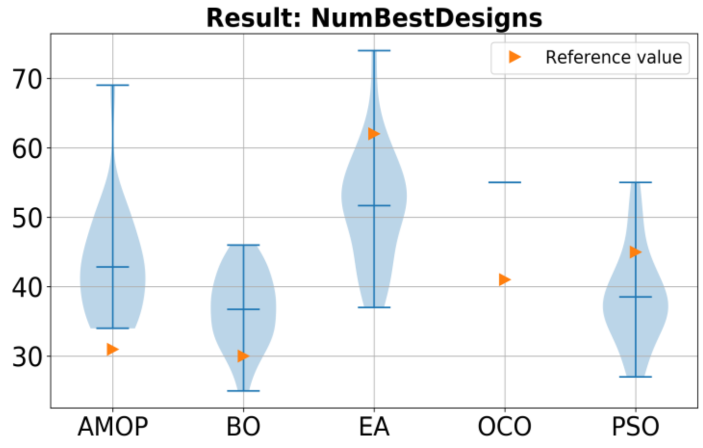
Examples for Multi-Objective-Optimization

eMachine: Results – setting 800_150_400_1000

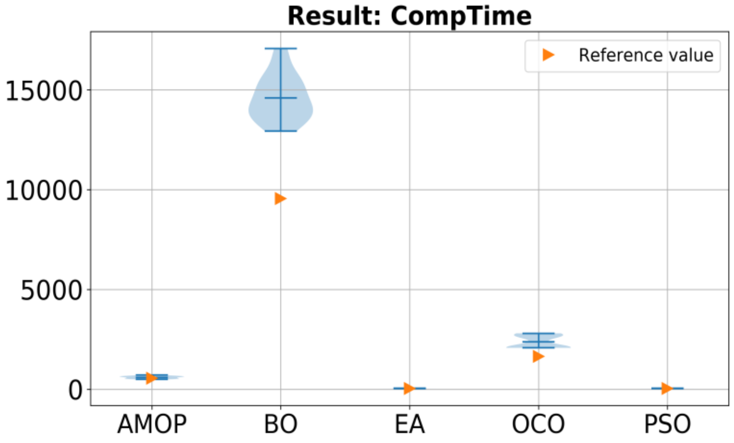
- BO shows the best results, followed by AMOP and OCO.
- The computing time for BO is quite high.
- The nature inspired algorithms EA shows better results than PSO.
- The number of best designs is high for EA.

- The “Reference value” belongs to a different setting !!!



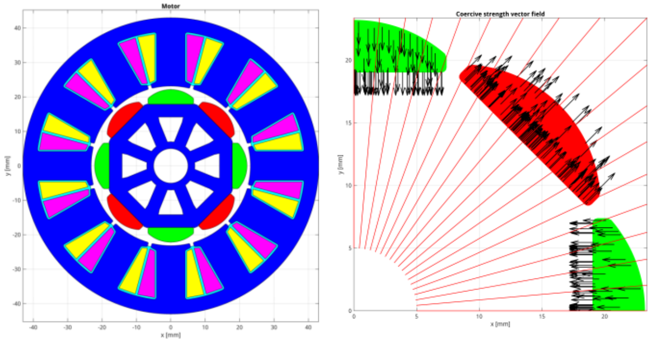


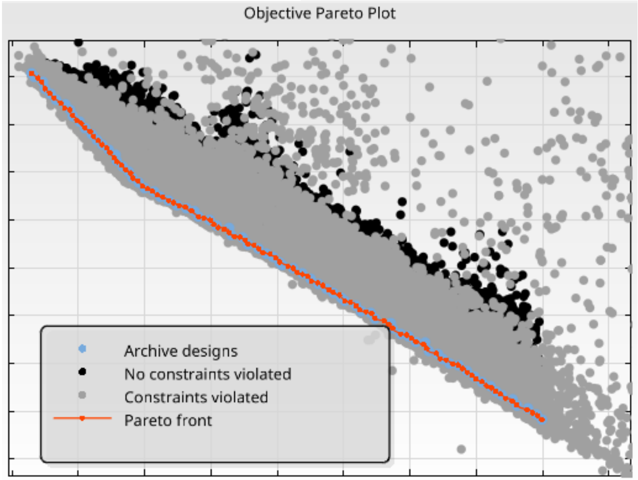
Ranking	
Algorithm	800/150/400/1000
BO	3
AMOP	7
OCO	9
EA	11
PSO	18



Examples for Multi-Objective-Optimization
Performance for an eDrive

- The geometry of the eDrive has 21 parameters. Four parameters have discrete values
  - Magnet_Material
  - Wire_Selector_X1
  - Do_Skew
  - Wire_Selector_X2
- The original optimization problem consists of 15 objective functions and 11 constraints, which was modified to 2 objective functions and 17 constraints.
- Because the calculation of the hypervolume does not allow negative values for the objective function which comes from the maximization of the torque (Trq_WP1), an offset of 7 was selected and a minimization of the difference to 7.
- 40 optimization runs were performed.





$Hypervolume: v = 13.497$

$Hypervolumereferenz:$

$obj\_cost\_indicator = 10$

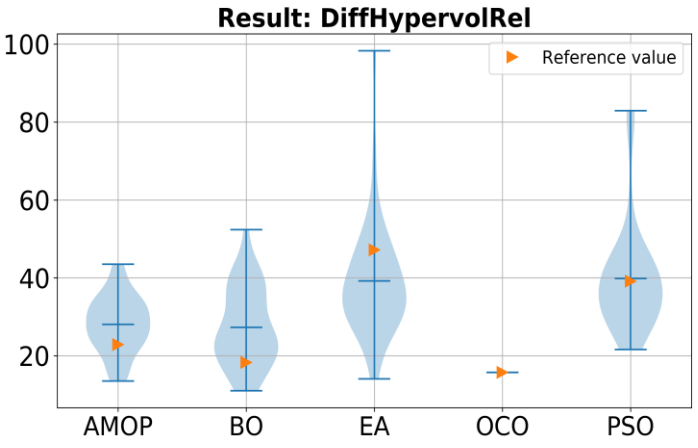
$obj\_Trq\_WP1 = 7$

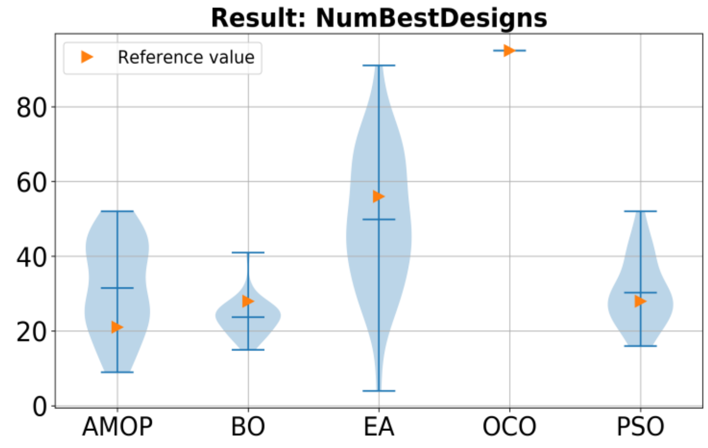
BOSCH

Examples for Multi-Objective-Optimization

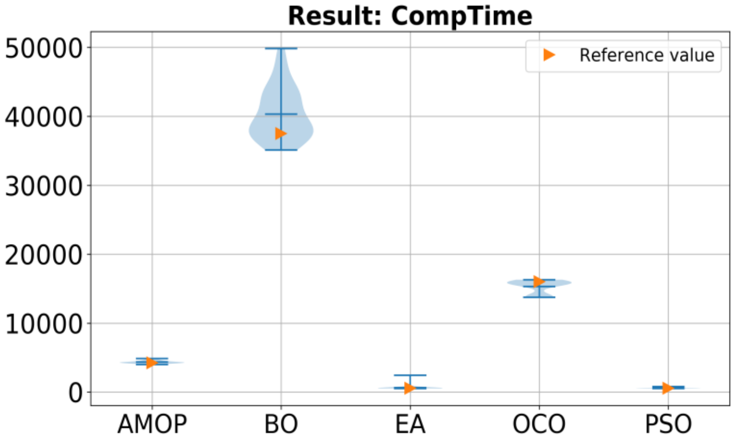
eDrive: Results – setting 1200/150/900/1500

- BO, OCO and AMOP have the same best ranking, because each one is the best in minimum value, mean value or maximum value of the difference to the hypervolume.
- The OCO has the highest number of best designs, but in relation to the maximum number of designs the BO has the highest number of best designs.
- The computation time of BO has the highest value, but OCO has a significant value.





Ranking	
Algorithm	1200/150/900/1500
BO	6
OCO	6
AMOP	7
EA	13
PSO	14

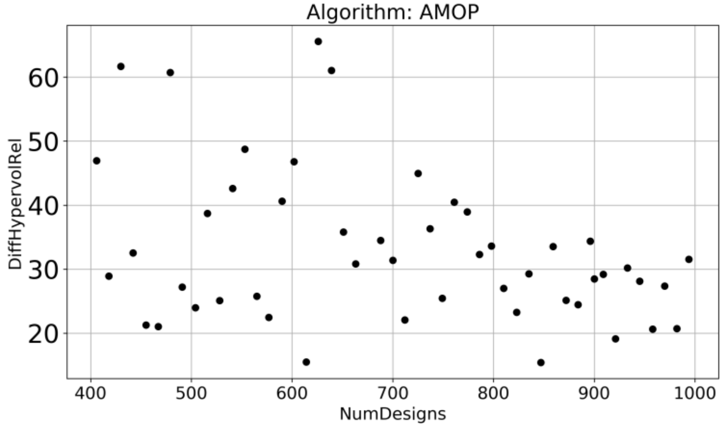


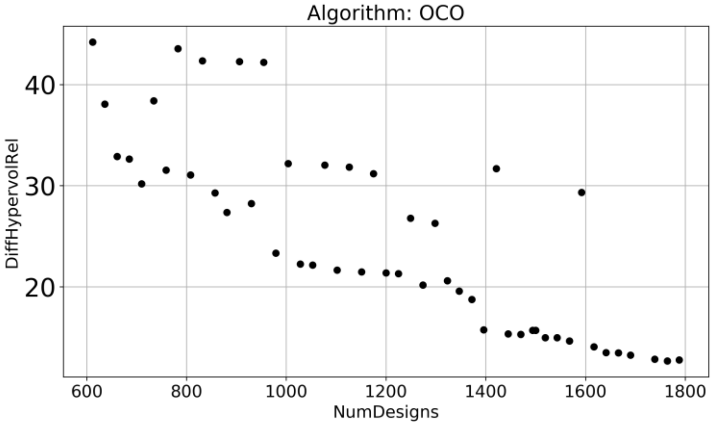
BOSCH

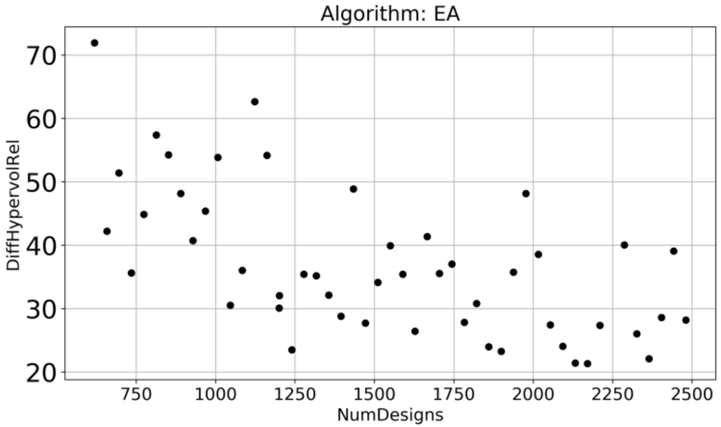
Examples for Multi-Objective-Optimization

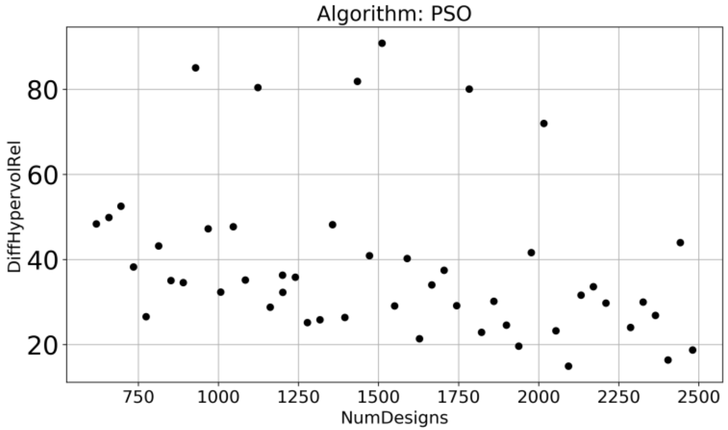
eDrive: Results – variation of the number of designs

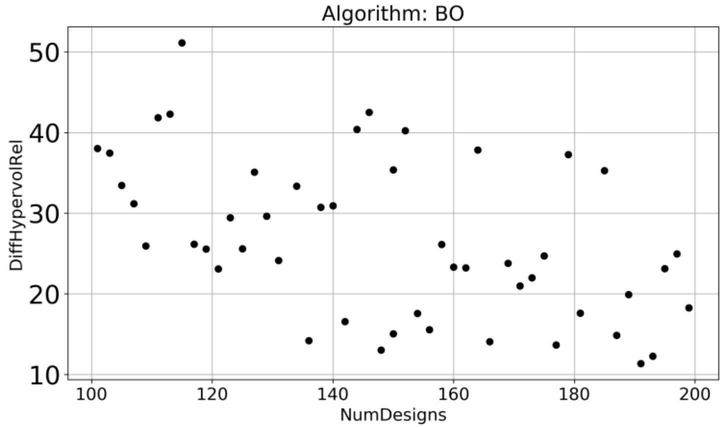
- There is no clear dependency between the relative difference to the reference Pareto Front and the number of designs.
- Sometimes a larger number of designs does not automatically lead to a better solution.











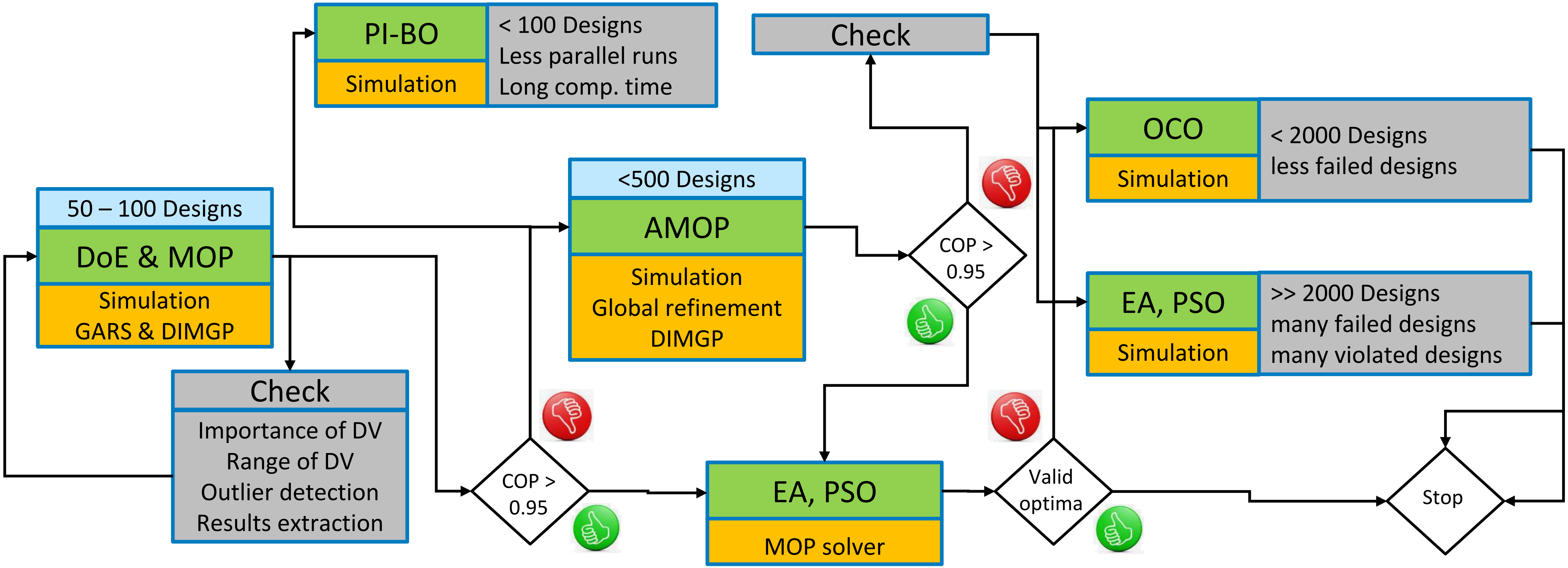
BOSCH

Benchmark of One-Click-Optimizer
Agenda

Benchmark of One-Click-Optimizer
Summary

- An automatic workflow could be established to benchmark different optimization algorithms.
- The integration of optiSLang-external algorithms is quite difficult. Several interfaces in Python were necessary to create the required files. Sometimes the OutputSlots like Ocriteria were used and sometimes the export of parameters/criteria via .csv format. It could be clarified whether a custom integration is a better approach.
- The adaptive and hybrid optimization algorithms showed the best performance. Often, the PI-BO showed the best results, but requires a long computation time. Perhaps the integration of PI-BO in optiSLang could be improved e.g. parallel training of criteria.
- The nature inspired optimization algorithms EA & PSO showed similar results, but they need much more designs for a good solution.
- The One Click Optimizer OCO does not show the best solution for all applications, but the OCO belongs to the better optimization algorithms.
- There are ideas to couple several methods sequentially to get better optimization results.

Benchmark of One-Click-Optimizer
Proposal for optimization process



CR/AME3 | 2023-06-22

BOSCH

WOST 2023
Benchmark of One-Click-Optimizer (OCO)

Discussion