On Automated Parameter Tuning, with Applications in Next-Generation Manufacturing

Lars Kotthoff and Patrick Johnson

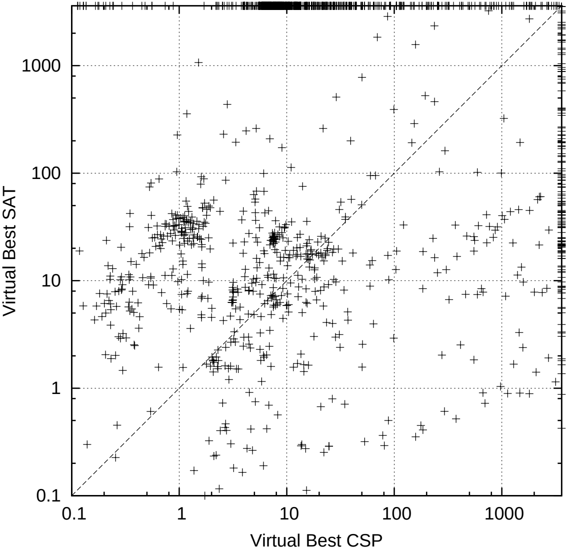
Artificially Intelligent Manufacturing Center
University of Wyoming
larsko,pjohns27@uwyo.edu

UCC, 02 April 2019

Big Picture

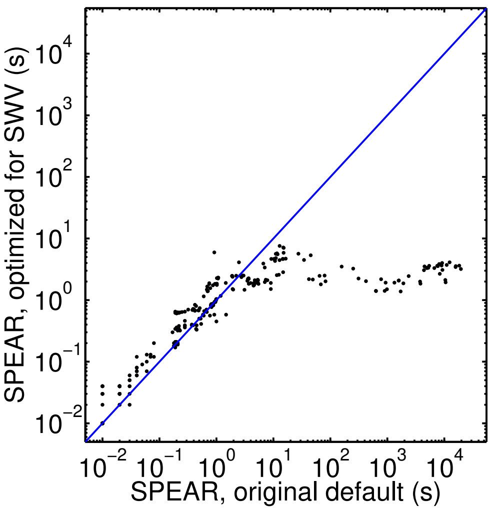
- advance the state of the art through meta-algorithmic techniques
- rather than inventing new things, use existing things more intelligently – automatically
- invent new things through combinations of existing things

Motivation – Performance Differences



Hurley, Barry, Lars Kotthoff, Yuri Malitsky, and Barry O'Sullivan. "Proteus: A Hierarchical Portfolio of Solvers and Transformations." In CPAIOR, 2014.

Motivation – Performance Improvements

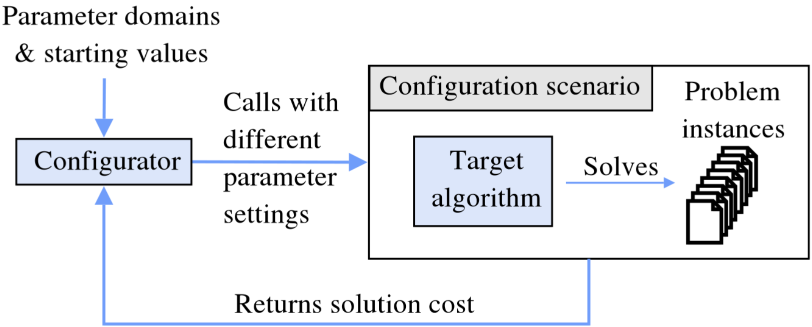


Hutter, Frank, Domagoj Babic, Holger H. Hoos, and Alan J. Hu. "Boosting Verification by Automatic Tuning of Decision Procedures." In FMCAD '07: Proceedings of the Formal Methods in Computer Aided Design, 27–34. Washington, DC, USA: IEEE Computer Society, 2007.

What to Tune – Parameters

- anything you can change that makes sense to change
- e.g. search heuristic, variable ordering, type of global constraint decomposition
- not random seed, whether to enable debugging, etc.
- some will affect performance, others will have no effect at all

Automated Parameter Tuning



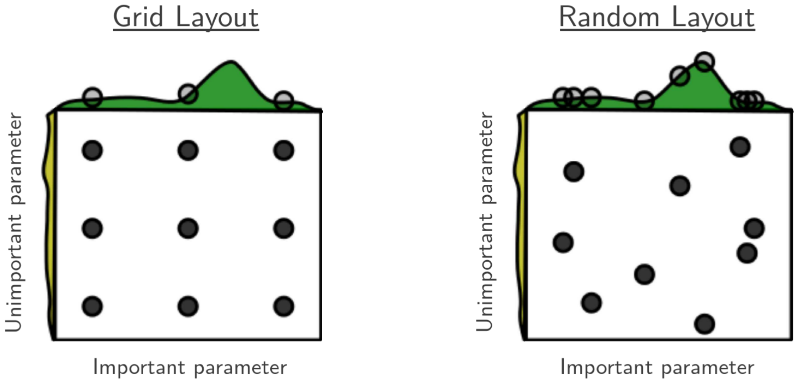
Frank Hutter and Marius Lindauer, “Algorithm Configuration: A Hands on Tutorial”, AAAI 2016

General Approach

- evaluate algorithm as black-box function
- observe effect of parameters without knowing the inner workings
- decide where to evaluate next
- balance diversification/exploration and intensification/exploitation
- repeat until satisfied

Grid and Random Search

- evaluate certain points in parameter space



Bergstra, James, and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization." J. Mach. Learn. Res. 13, no. 1 (February 2012): 281–305.

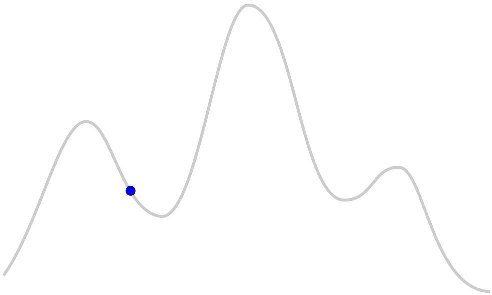
Local Search

- start with random configuration
- change a single parameter (local search step)
- if better, keep the change, else revert
- repeat, stop when resources exhausted or desired solution quality achieved
- restart occasionally with new random configurations

Local Search Example

graphics by Holger Hoos

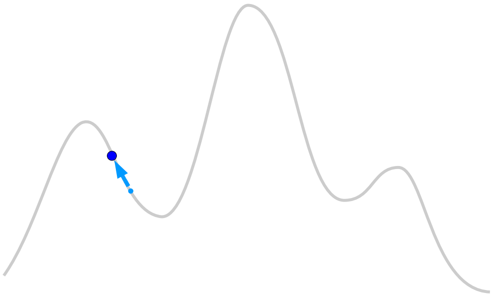
Local Search Example



Initialisation

graphics by Holger Hoos

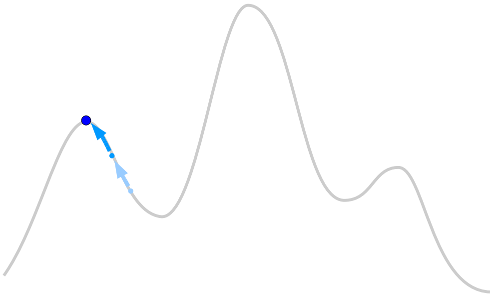
Local Search Example



Local Search

graphics by Holger Hoos

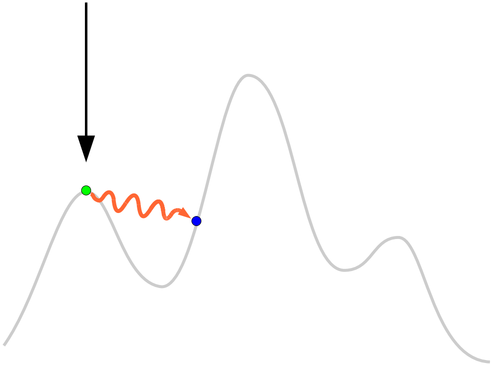
Local Search Example



Local Search

graphics by Holger Hoos

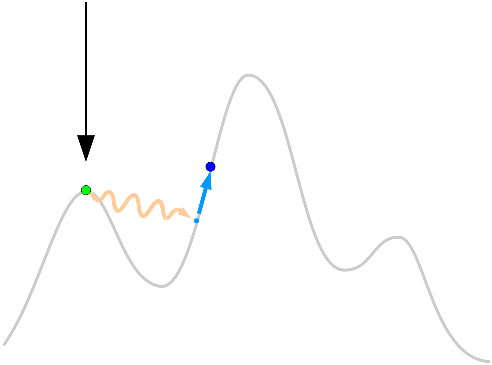
Local Search Example



Perturbation

graphics by Holger Hoos

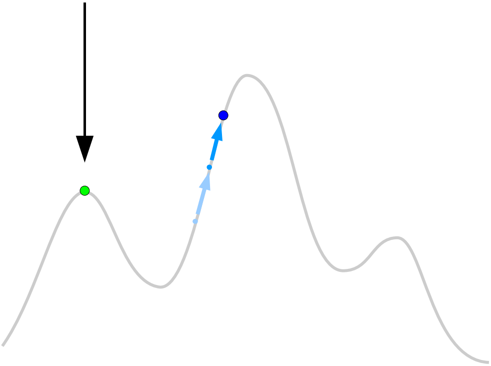
Local Search Example



Local Search

graphics by Holger Hoos

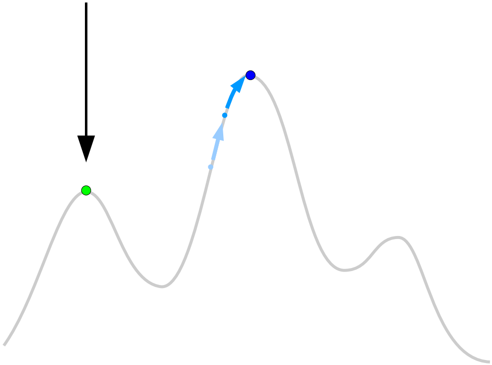
Local Search Example



Local Search

graphics by Holger Hoos

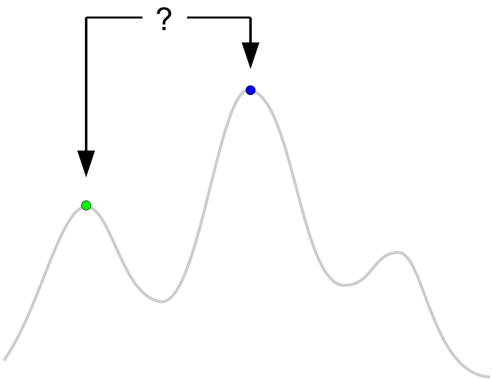
Local Search Example



Local Search

graphics by Holger Hoos

Local Search Example



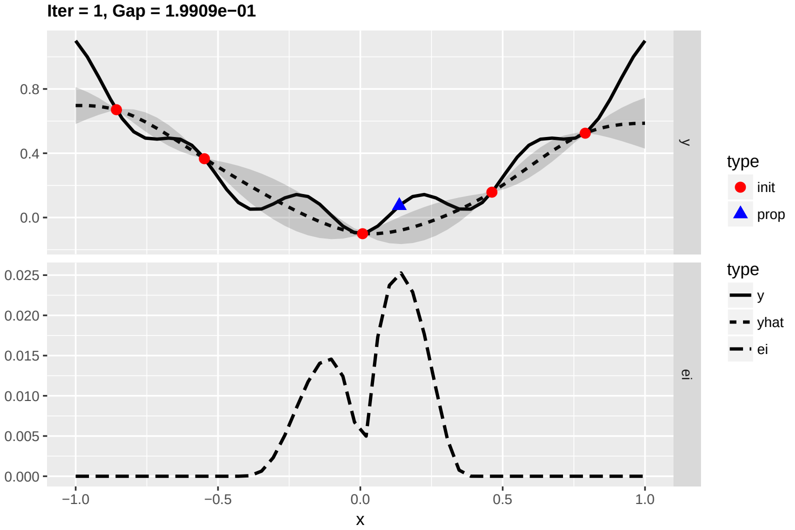
Selection (using Acceptance Criterion)

graphics by Holger Hoos

Surrogate-Model-Based Search

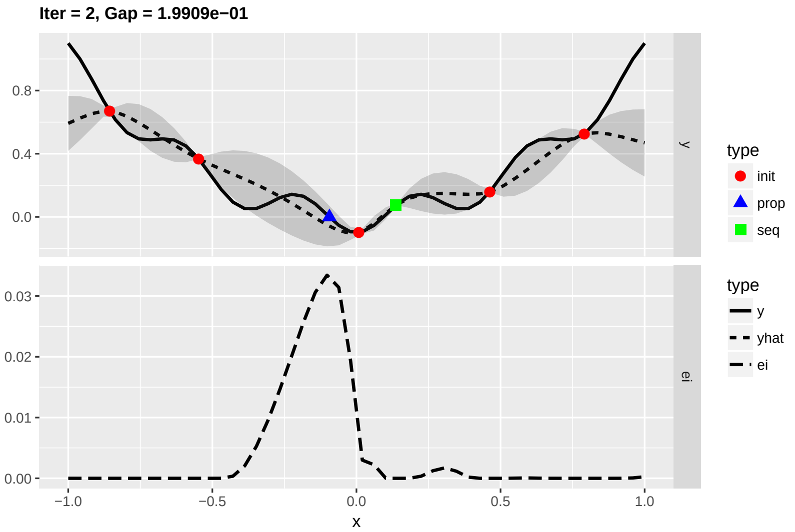
- evaluate small number of initial (random) configurations
- build surrogate model of parameter-performance surface based on this
- use model to predict where to evaluate next
- repeat, stop when resources exhausted or desired solution quality achieved
- allows targeted exploration of promising configurations

Surrogate-Model-Based Search Example



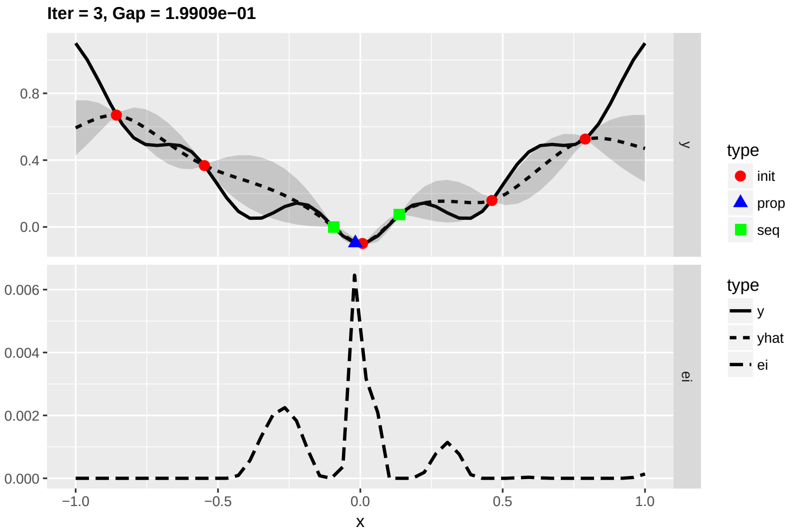
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



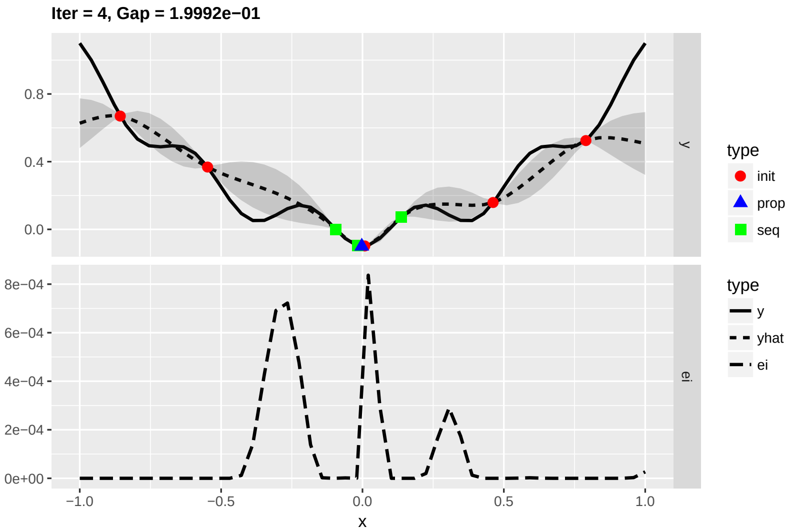
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



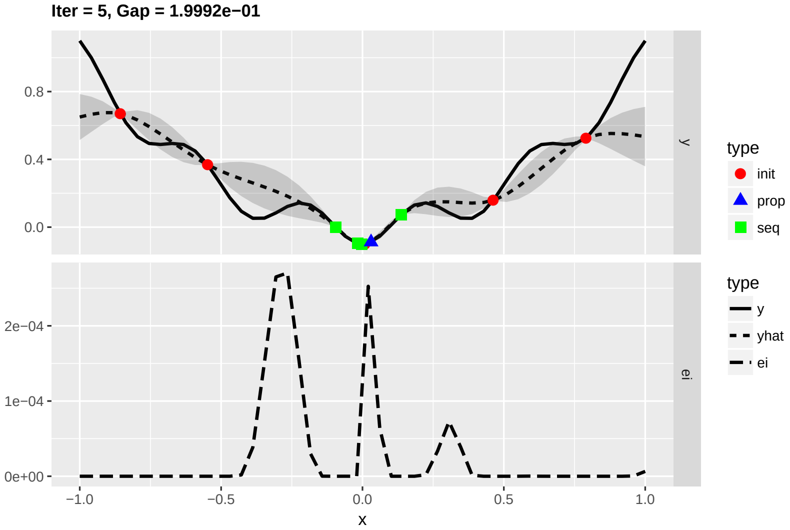
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



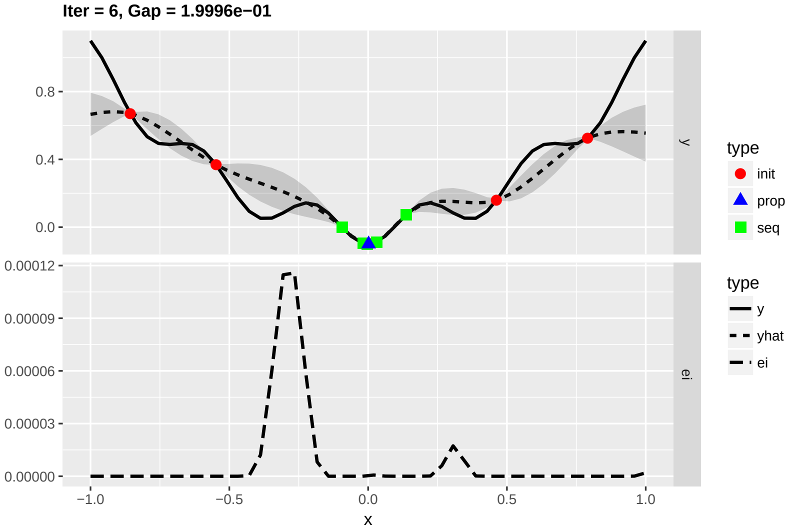
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



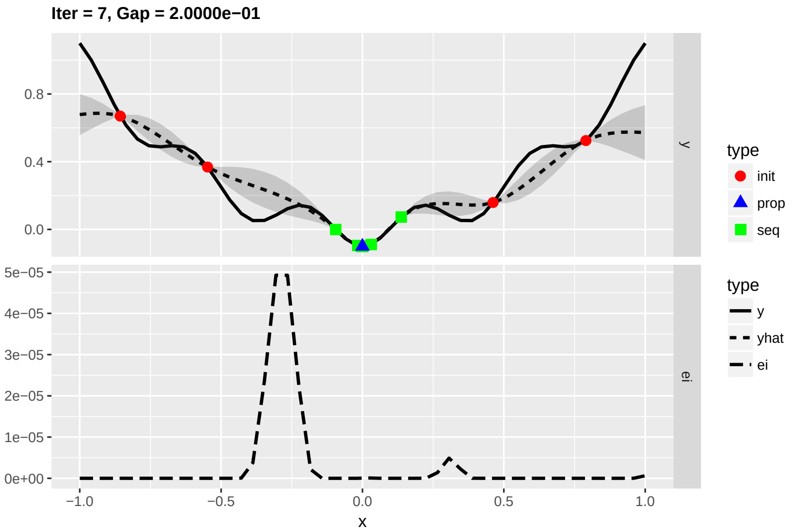
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



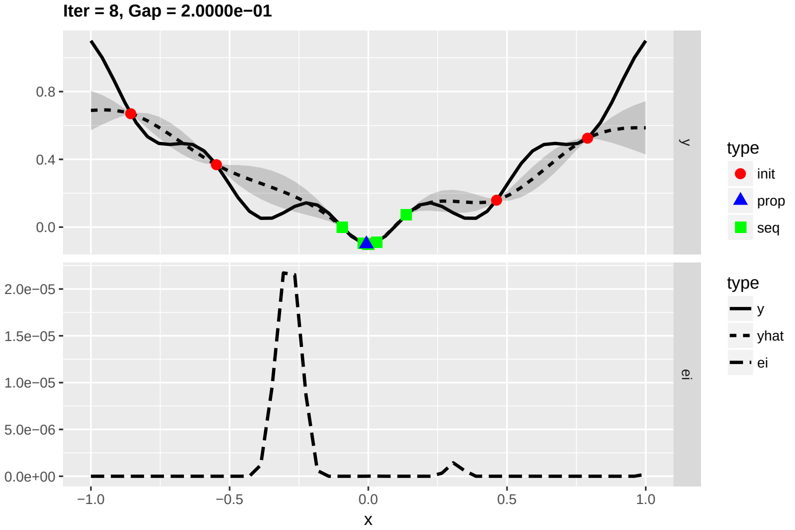
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



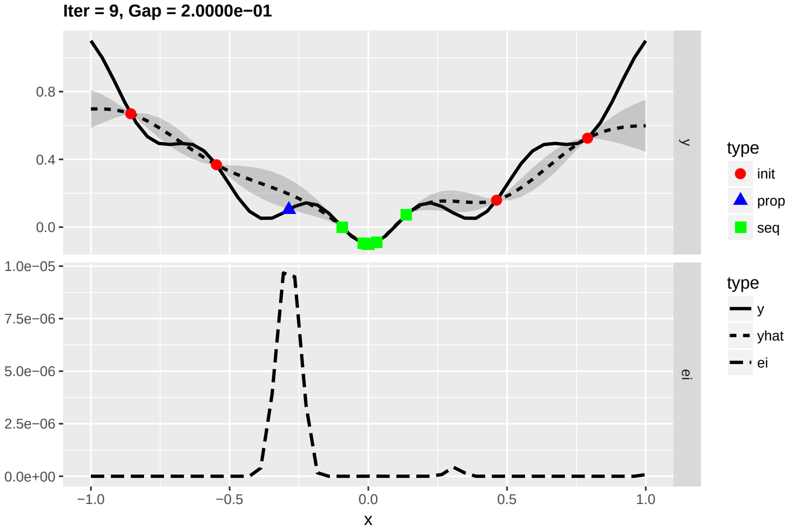
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



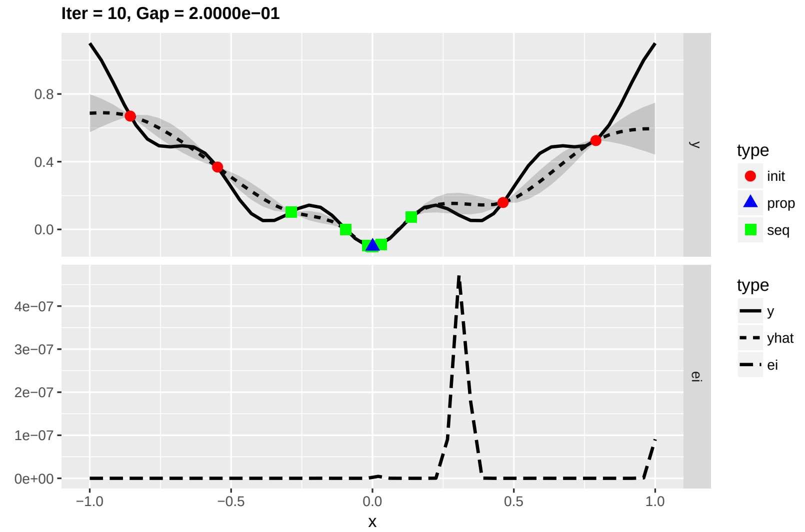
Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Surrogate-Model-Based Search Example



Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. "MlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions," March 9, 2017. http://arxiv.org/abs/1703.03373.

Two-Slide MBO

```
# http://www.cs.uwyo.edu/~larsko/mbo.py
params = { 'C': np.logspace(-2, 10, 13),
           'gamma': np.logspace(-9, 3, 13) }
param_grid = [ { 'C': x, 'gamma': y } for x in params['C']
                                       for y in params['gamma'] ]
# [{'C': 0.01, 'gamma': 1e-09}, {'C': 0.01, 'gamma': 1e-08}...]

initial_samples = 3
evals = 10
random.seed(1)

def est_acc(pars):
    clf = svm.SVC(**pars)
    return np.median(cross_val_score(clf, iris.data, iris.target, cv = 10))

data = []
for pars in random.sample(param_grid, initial_samples):
    acc = est_acc(pars)
    data += [ list(pars.values()) + [ acc ] ]
# [[1.0, 0.1, 1.0],
#  [1000000000.0, 1e-07, 1.0],
#  [0. 1, 1e-06,0.9333333333333333]]
```

Two-Slide MBO

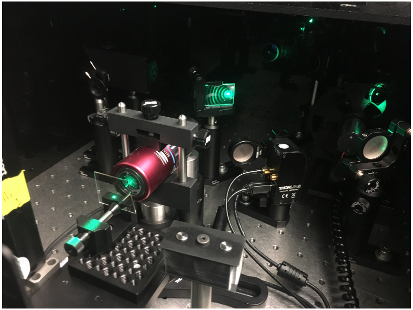
```
regr = RandomForestRegressor(random_state = 0)
for evals in range(0, evals):
    df = np.array(data)
    regr.fit(df[:,0:2], df[:,2])

    preds = regr.predict([ list(pars.values()) for pars in param_grid ])
    i = preds.argmax()

    acc = est_acc(param_grid[i])
    data += [ list(param_grid[i].values()) + [ acc ] ]
    print("{}: best predicted {} for {}, actual {}"
            .format(evals, round(preds[i], 2), param_grid[i], round(acc, 2)))

i = np.array(data)[:,2].argmax()
print("Best accuracy ({}) for parameters {}".format(data[i][2], data[i][0:2]))
```

Two-Slide MBO (slide 3)

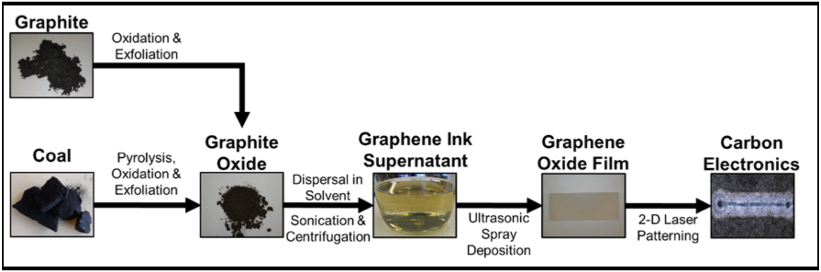
```
0: best predicted 0.99 for {'C': 1.0, 'gamma': 1e-09}, actual 0.93
1: best predicted 0.99 for {'C': 1000000000.0, 'gamma': 1e-09}, actual 0.93
2: best predicted 0.99 for {'C': 1000000000.0, 'gamma': 0.1}, actual 0.93
3: best predicted 0.97 for {'C': 1.0, 'gamma': 0.1}, actual 1.0
4: best predicted 0.99 for {'C': 1.0, 'gamma': 0.1}, actual 1.0
5: best predicted 1.0 for {'C': 1.0, 'gamma': 0.1}, actual 1.0
6: best predicted 1.0 for {'C': 1.0, 'gamma': 0.1}, actual 1.0
7: best predicted 1.0 for {'C': 1.0, 'gamma': 0.1}, actual 1.0
8: best predicted 1.0 for {'C': 0.01, 'gamma': 0.1}, actual 0.93
9: best predicted 1.0 for {'C': 1.0, 'gamma': 0.1}, actual 1.0
Best accuracy (1.0) for parameters [1.0, 0.1]
```

Application – Optimizing Graphene Oxide Reduction

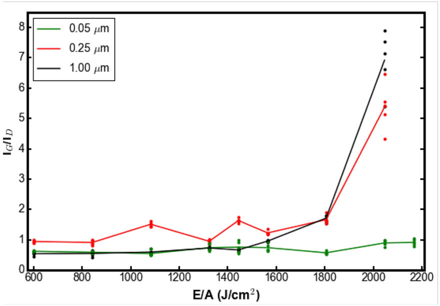
- reduce graphene oxide to graphene through laser irradiation
- allows to create electrically conductive lines in insulating material
- laser parameters need to be tuned carefully to achieve good results

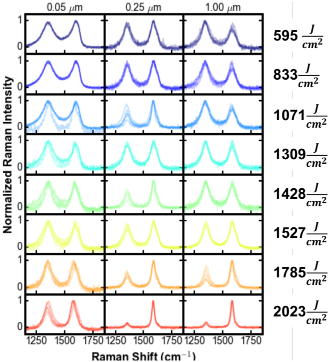


From Graphite/Coal to Carbon Electronics

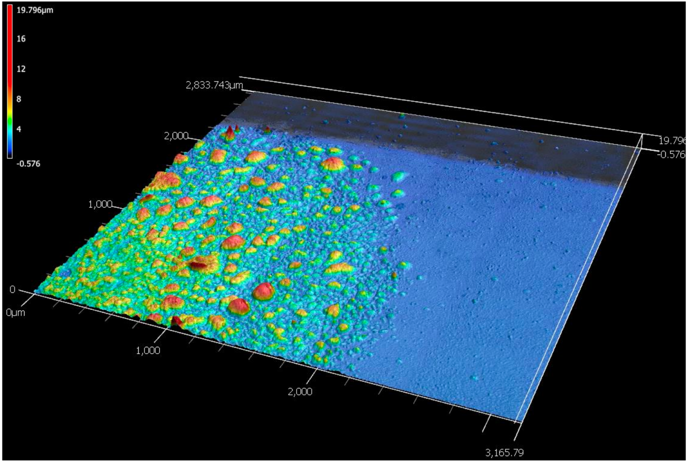


Evaluation of Irradiated Material

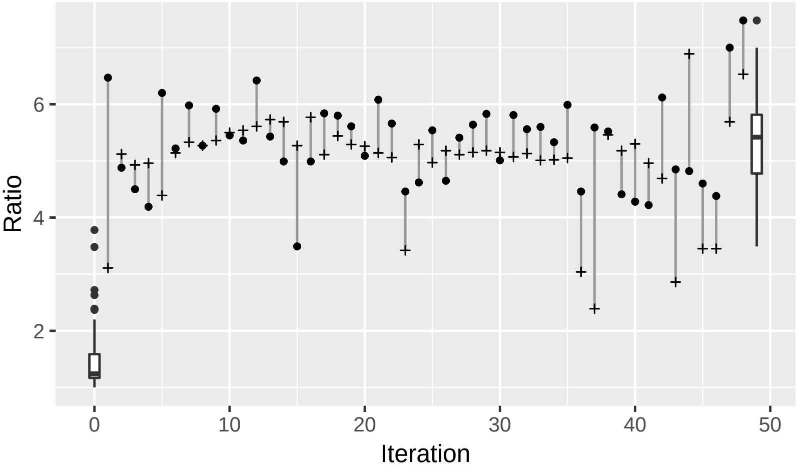




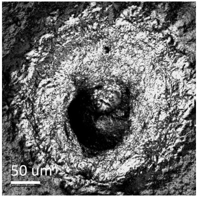
Morphology of Irradiated Material



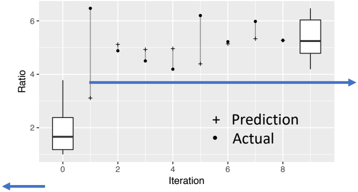
Surrogate-Model-Based Optimization

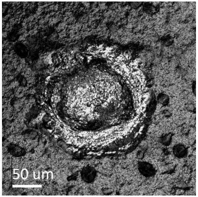


Surrogate-Model-Based Optimization



During Training

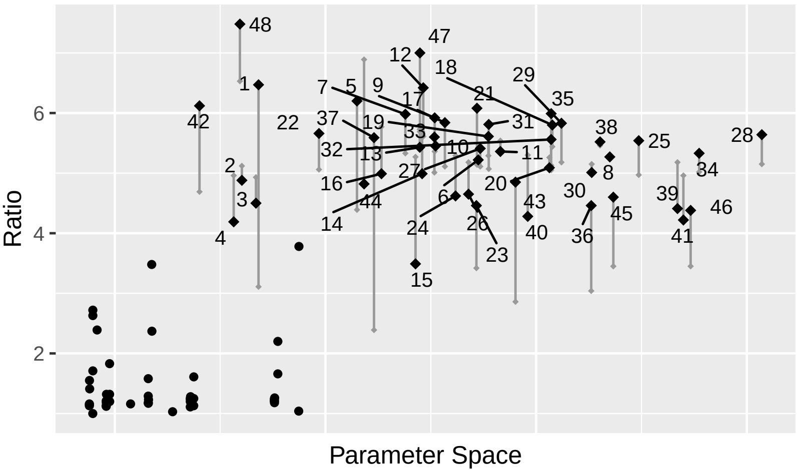




After 1st prediction

- Predictions work even with small training dataset (19 points)
- AI Model achieved I_G/I_D ratio (>6) after 1st prediction

Explored Parameter Space



Tools and Resources

iRace	http://iridia.ulb.ac.be/irace/
TPOT	https://github.com/EpistasisLab/tpot
mlrMBO	https://github.com/mlr-org/mlrMBO
SMAC	http://www.cs.ubc.ca/labs/beta/Projects/SMAC/
Spearmint	https://github.com/HIPS/Spearmint
TPE	https://jaberg.github.io/hyperopt/

COSEAL group for COnfiguration and SElection of ALgorithms: https://www.coseal.net/

Out soon: edited book on automated machine learning https://www.automl.org/book/ (Frank Hutter, Lars Kotthoff, Joaquin Vanschoren)

More on our applications: https://www.uwyo.edu/ceas/engineering-initiative/aim/

We're hiring!

Several funded positions available.

COLLEGE OF ENGINEERING & APPLIED SCIENCE
UNIVERSITY OF WYOMING

WYOMING NASA SPACE GRANT CONSORTIUM

SCHOOL OF ENERGY RESOURCES
UNIVERSITY OF WYOMING

NSF