Automated Experimentation for Online Learning and Adaptation

Ilias Gerostathopoulos

Software & Systems Engineering
Fakulty of Informatics
Technical University Munich

www4.in.tum.de/

About me

Engineering Diploma (Master's equivalent)
in Electrical and Computer Engineering

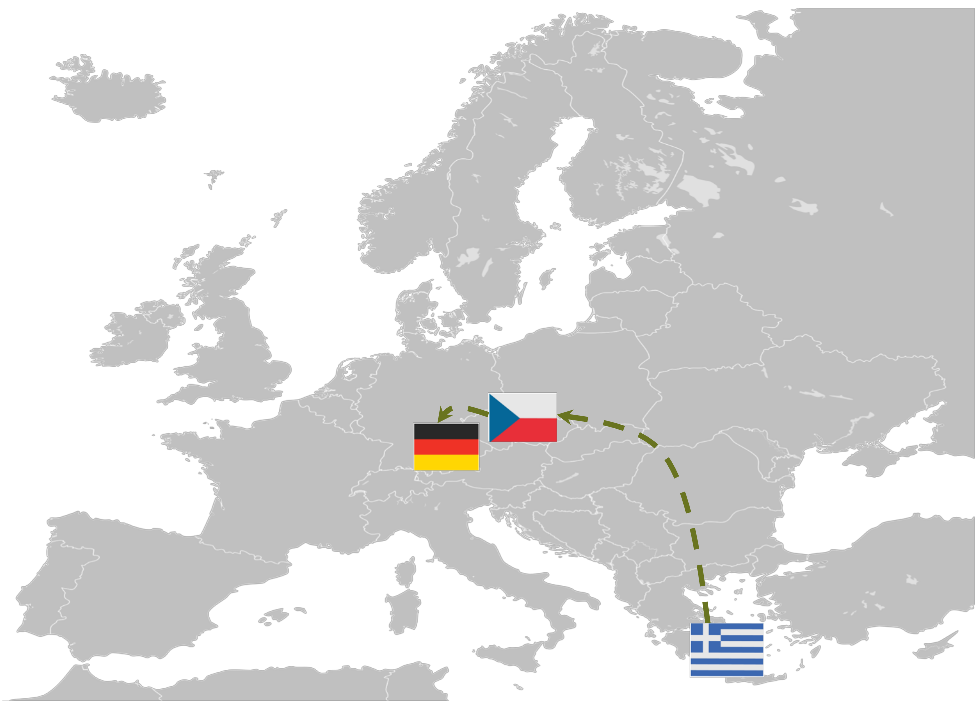
1987

2010

Sep 2015

now

* Kos, Greece



Doctorate in
Computer Science

Postdoctoral
Researcher

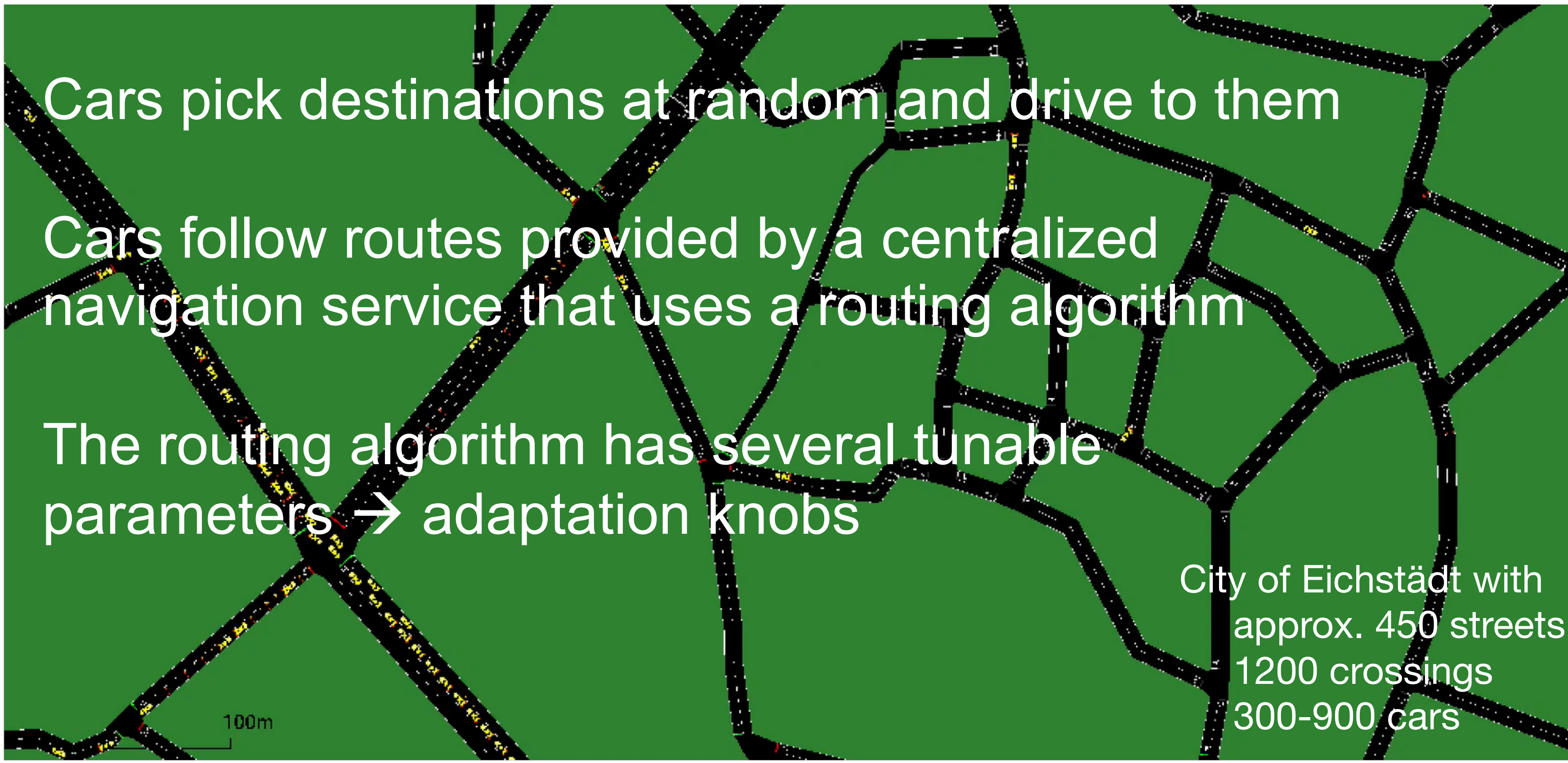
Explainability in Software Systems

Who should the systems be explainable to?

- **Users** → **interface design, HCI**
- **Developers** → **debugging, verification**
- **Operators** → **server logging, etc.**
- **Architects & Managers** → **online experimentation**
- …

Some systems are hard to optimize at runtime

CrowdNav: crowdsourced traffic navigation service (~Waze)



Cars pick destinations at random and drive to them

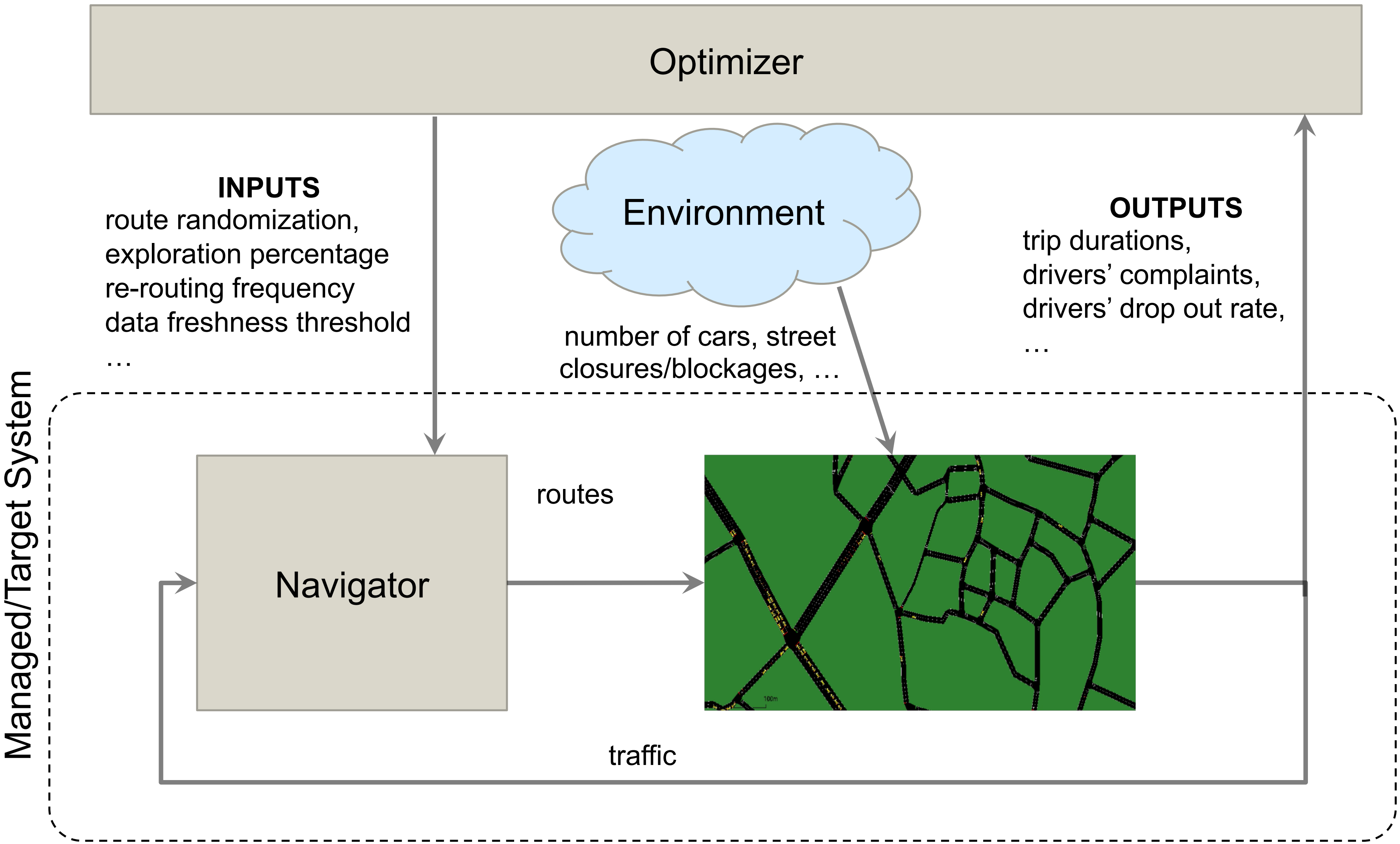
Cars follow routes provided by a centralized navigation service that uses a routing algorithm

The routing algorithm has several tunable parameters → adaptation knobs

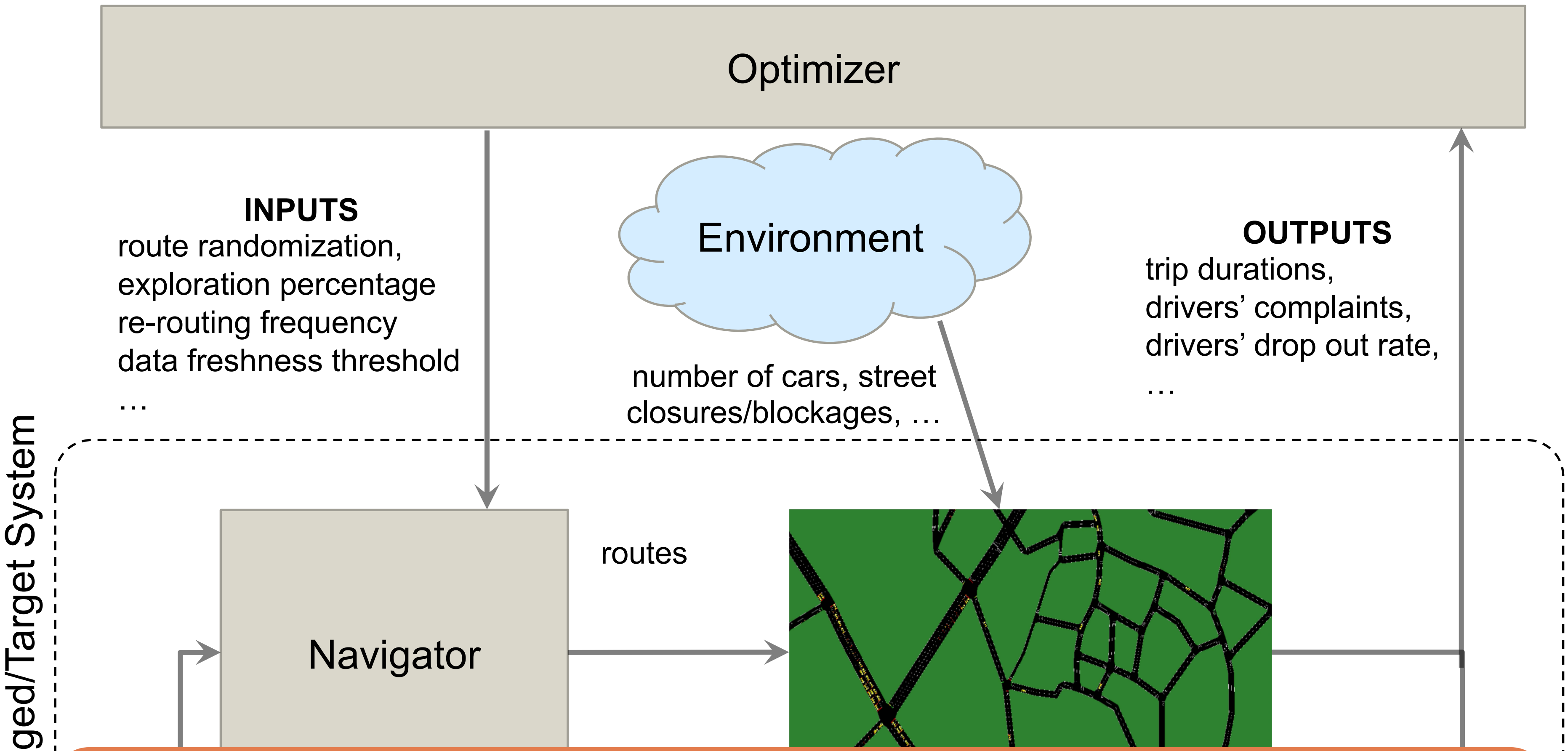
City of Eichstädt with
- approx. 450 streets
- 1200 crossings
- 300-900 cars

More at https://www.hpi.uni-potsdam.de/giese/public/selfadapt/exemplars/

Self-adaptation in CrowdNav



Self-adaptation in CrowdNav



Goal: Minimize trip durations while keeping complaints at regular rate, considering environment and system changes

Some systems are hard to optimize at runtime—why?

(1) They have to be treated as **black-boxes**

Difficult to obtain fine-grained models of internal system behavior

Think of creating (and updating at runtime) a state chart for a fleet of cars and their interactions and complex dependencies

(2) Their output is highly **stochastic**

Cannot rely on single measurements but need to cope with variance by collecting many measurements and carefully choosing which metric to use

Ideas behind Automated Online Experimentation

(1) They have to be treated as **black-boxes**

Rely only on the essential input and output system parameters: **what can be monitored, what can be changed**

(2) Their output is highly **stochastic**

Apply **statistical methods** to compare the output of the system under different configurations

How to select a configuration to try out (an experiment)?

Option 1

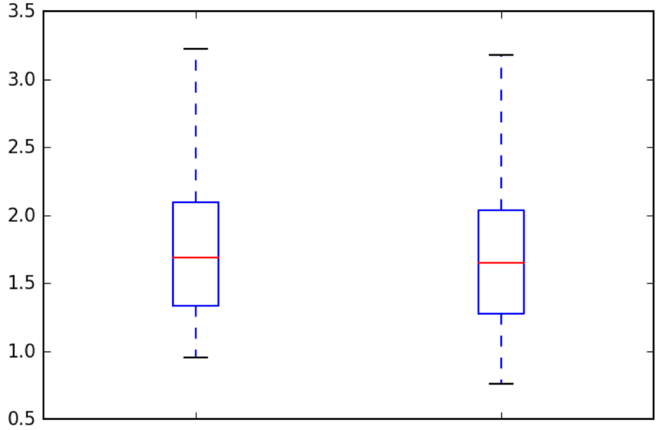
Manually select a promising configuration (other than the baseline)

This is what is mostly done in online experimentation platforms that perform A/B testing!

- Microsoft
- Google
- Netflix
- …

Apply selected configuration

Result of configuration



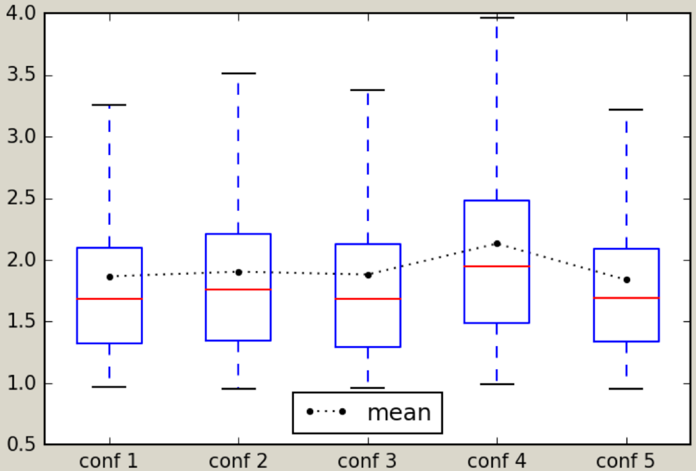
T-test to compare to outputs of baseline configuration

[3]

How to select a configuration to try out (an experiment)?

Option 2

Automate the selection with an optimization/search algorithm

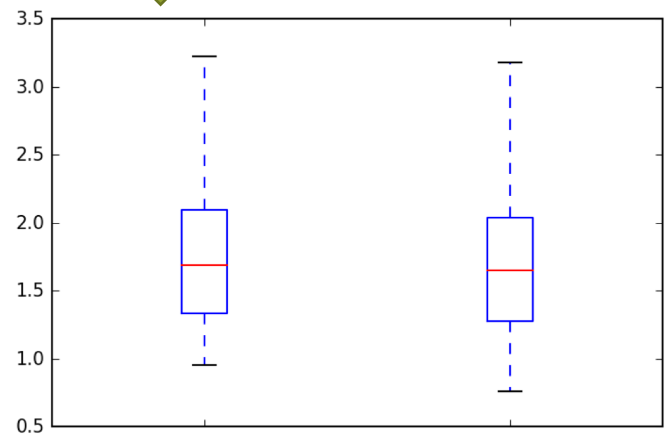


Bayesian optimization with Gaussian Processes

Apply selected configuration

System output values

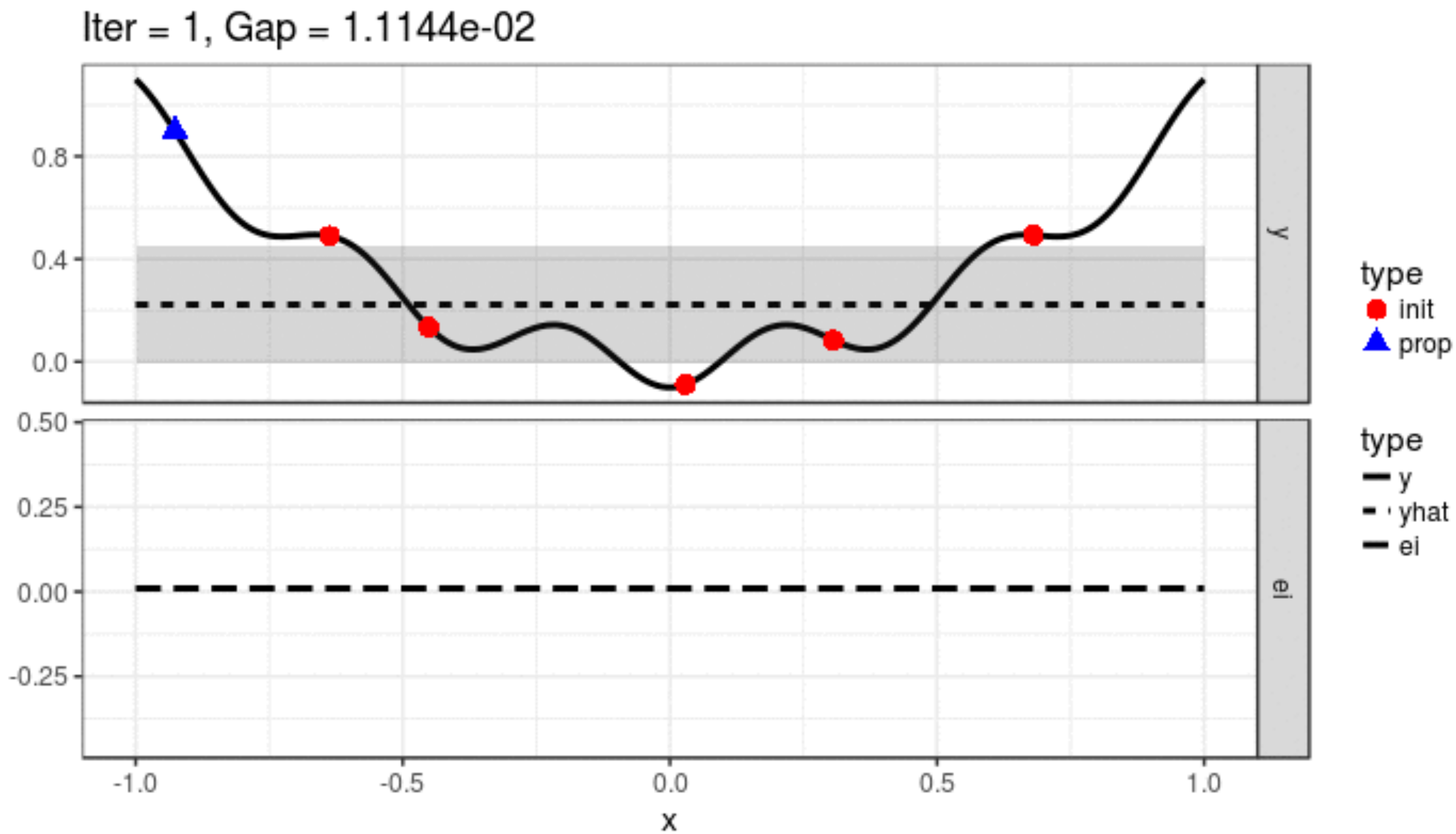
Result of BEST configuration



T-test to compare to outputs of baseline configuration

[3]

Detour: Bayesian Optimization



How to select a configuration to try out (an experiment)?

Option 3

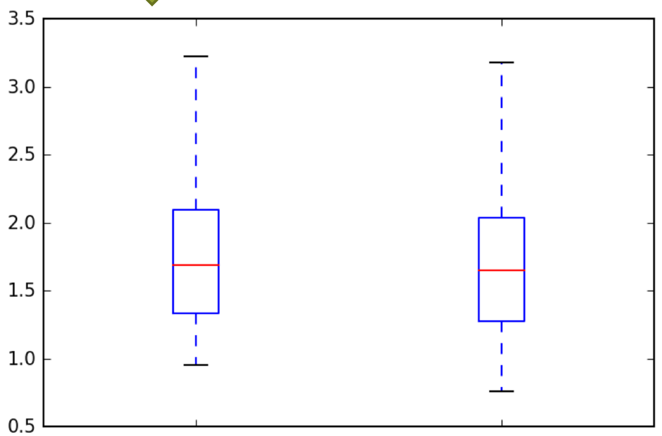
First, select a **subset** of the parameters that have **impact on the output**

Then, use **only these parameters** in the optimization algorithm

Apply selected configuration

System output values

Result of BEST configuration

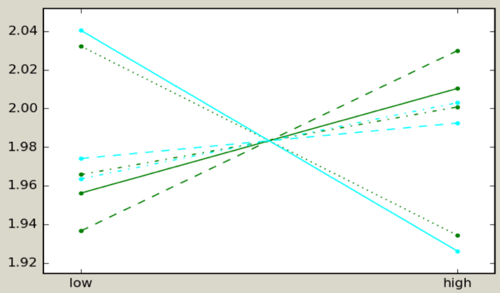


T-test to compare to outputs of baseline configuration

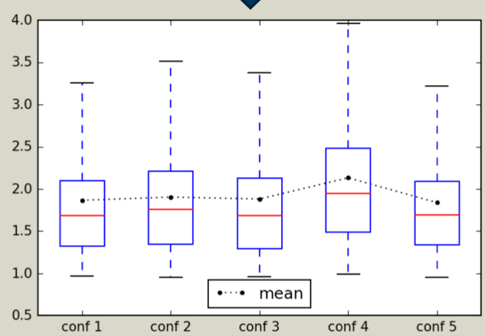
[3]

How to select a configuration to try out (an experiment)?

Option 3



Factorial ANOVA (Grid search)

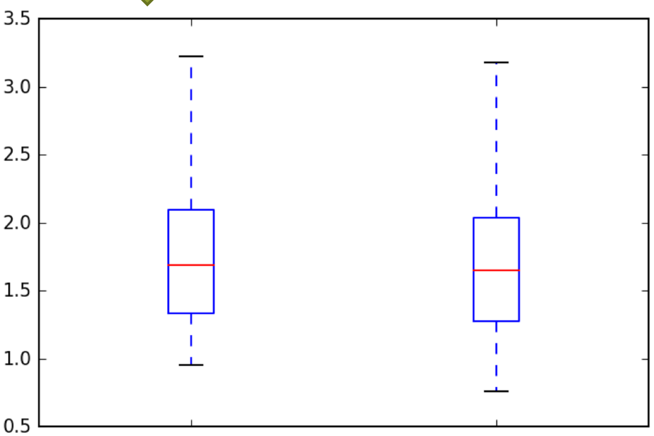


Bayesian optimization with Gaussian Processes

Apply selected configuration

System output values

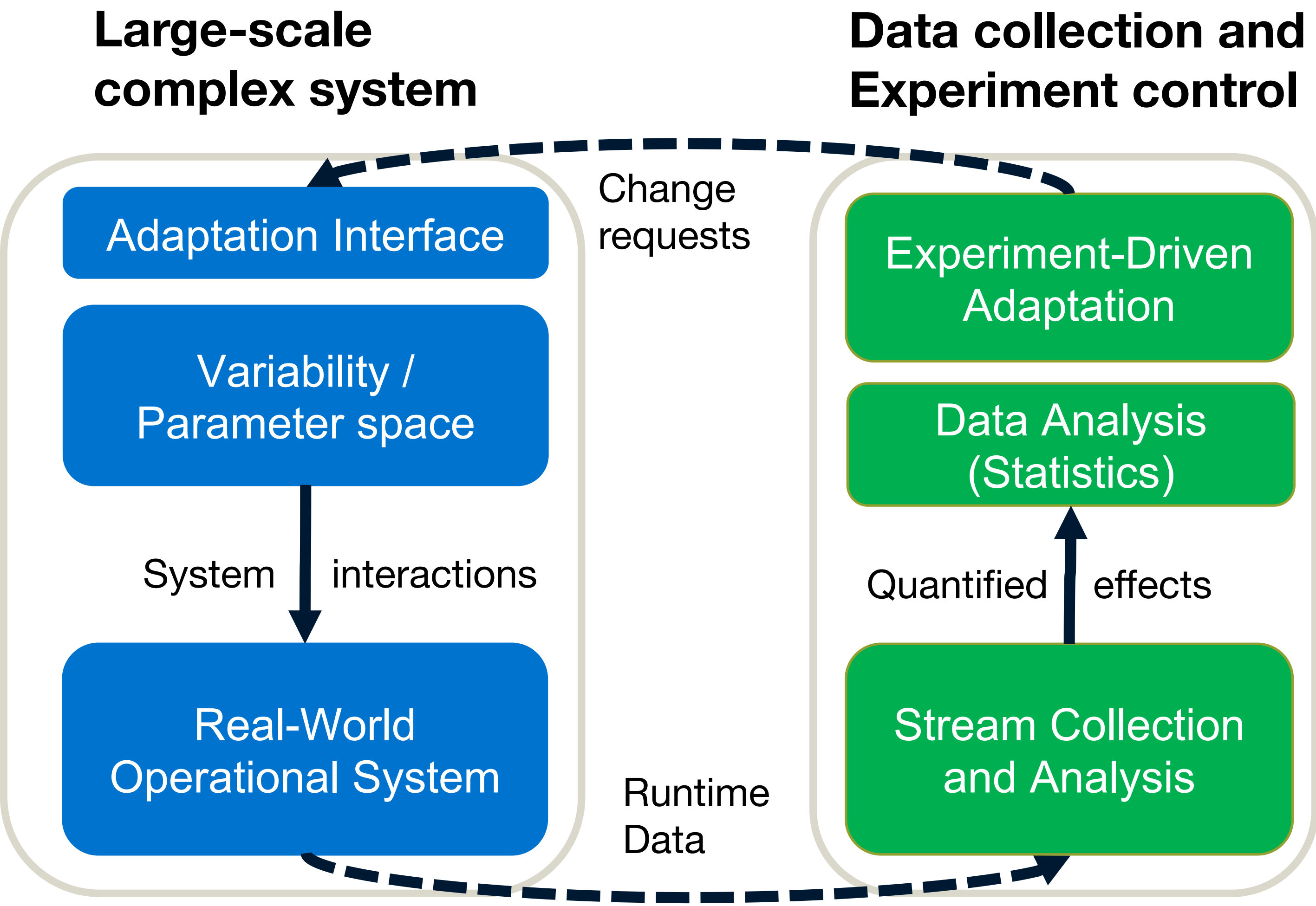
Result of BEST configuration



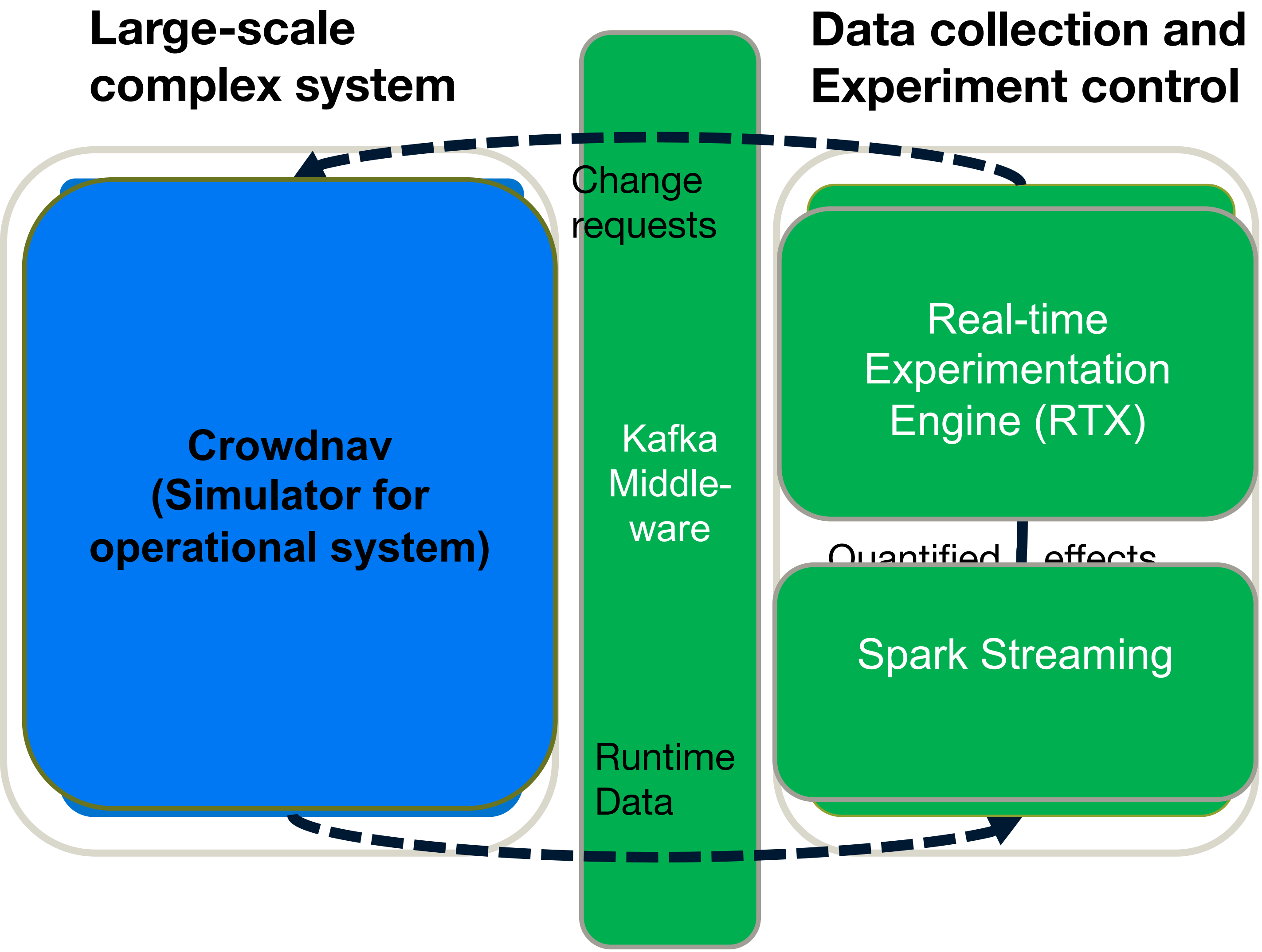
T-test to compare to outputs of baseline configuration

[3]

High-level model for Automated Online Experimentation



Tools for Automated Online Experimentation



Automated Online Experimentation: Challenges and Results

Running experiments comes with cost →

*Challenge: Find a **optimal solution** at **optimal cost***

Different costs of running experiments in production:

1. Cost of **running experiments** w.r.t. time, operational cost and opportunity cost
2. Cost of **system adaptation**, e.g. change the number of drivers
3. Cost of running **“bad” configurations**, e.g. negative user experience

Lack of integrated approach for experiments

Most research in data analysis & optimization focuses on 1.

Work on A/B testing also on 3.

Example Scenario and Experimentation Strategies

Focus on trips between two city areas

- E.g. residential and business areas
- 300 cars commute between areas
- 300 random cars

Selected parameters for routing

- **Route randomization**: amount of randomization among similar routes
- **Data freshness threshold**: how long old data is used

Experiment Selection Strategies

A. Bayesian optimization with Gaussian processes

Balances between exploration of unknown areas vs promising areas

B. Full factorial design

Systematic, extensive exploration of all options

C. Local search starting from a previously known stable configuration

Cautious approach, assuming a stable starting configuration

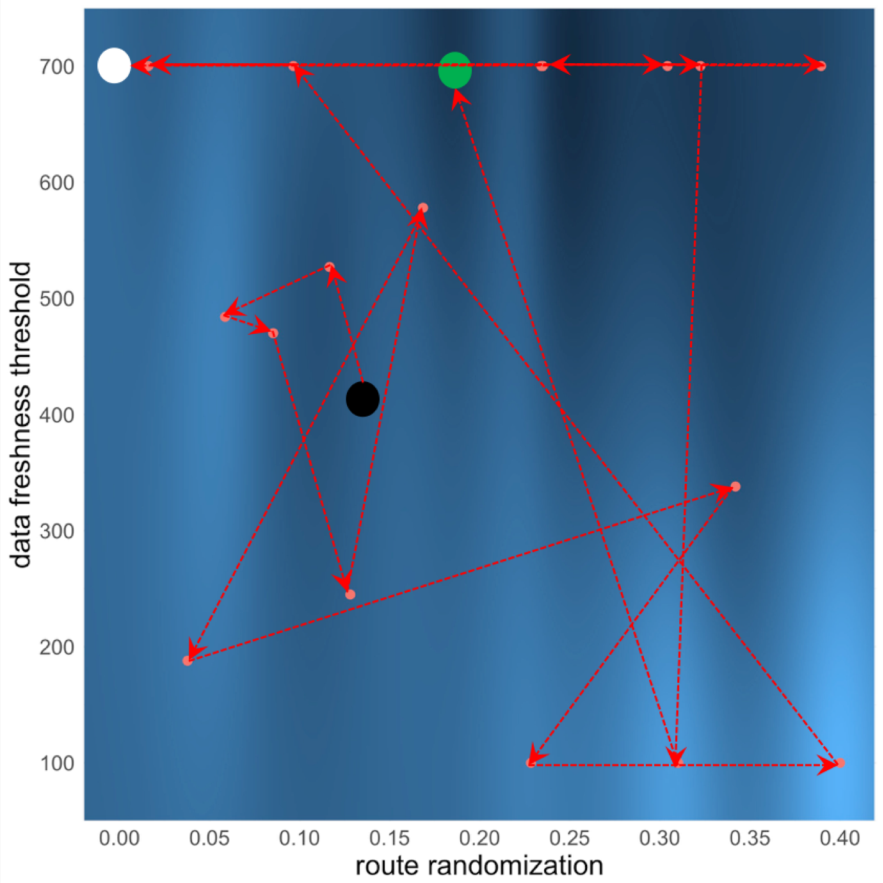
Locally looks for promising areas (similar to „gradient descent“)

Cost model: Number of **customers complaints**, issued if a ride took longer than expected

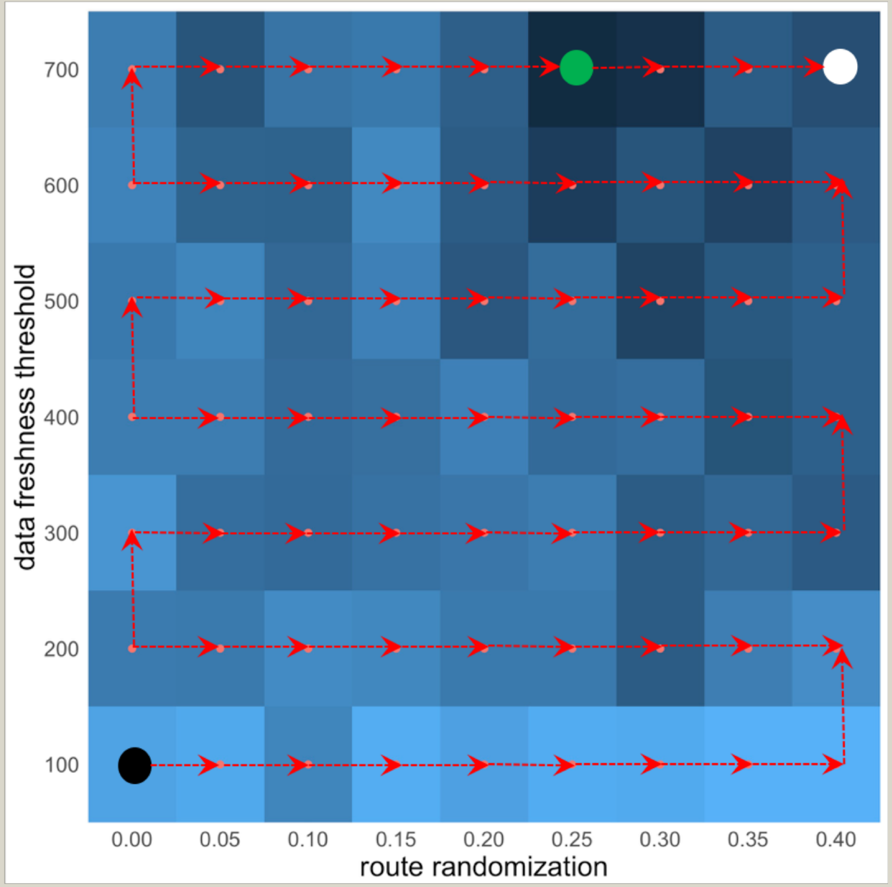
- Fixed percentage of trip overhead leads to “complaint”, due to longer routes and traffic jams
- Consider operation of service and effort for parameter change as fixed

Joint work with Janek Thomas & Bernd Bischl, Ludwig Maximilian University of Munich

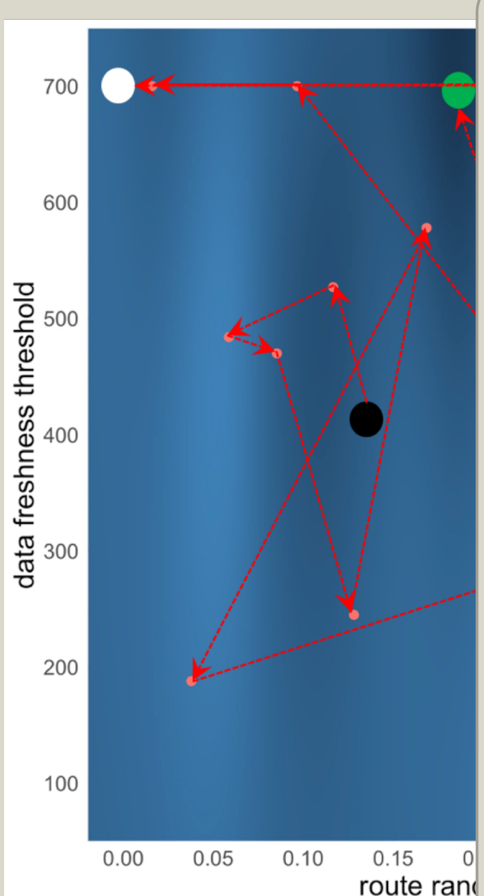
Evaluation Results



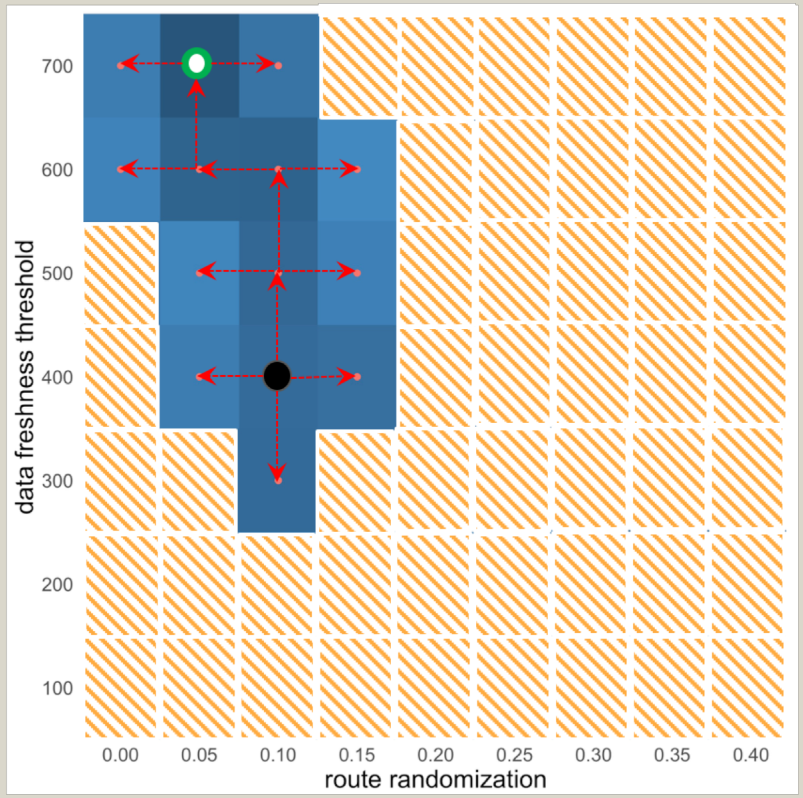
A) BO with Gaussian Processes
Type #3 cost : 4176 complaints



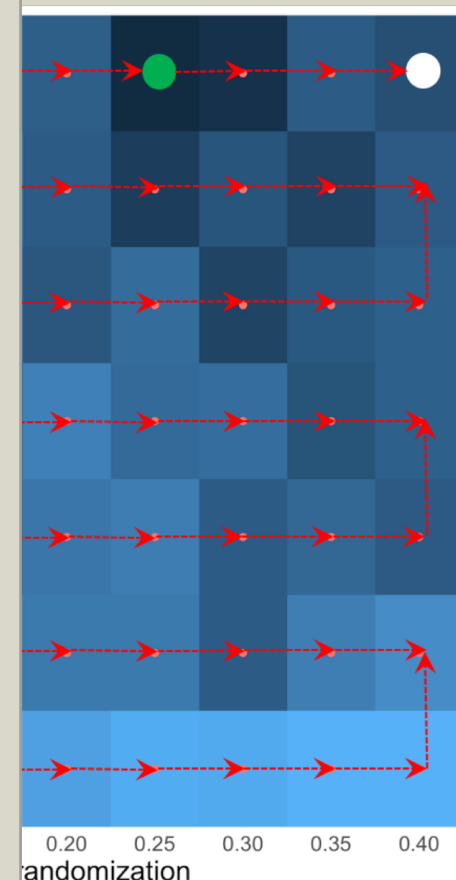
B) Factorial Design
Type #3 cost : 15.405 complaints



A) BO with Gaussian Processes
Type #3 cost : 4176 complaints



C) Local Search
Type #3 cost: 3226 complaints



B) Factorial Design
Type #3 cost : 15.405 complaints

Summary

Making complex, open ended systems self-adaptive is a challenge

Statistical guarantees should be sought after

Automated Online Experimentation can help in deriving insights directly from data

Different experimentation strategies yield different types of costs, also in terms of customer dissatisfaction

https://github.com/iliasger/OEDA

GitHub

http://www.continuous-se.org

RCoSE/DDrEE 2019

Joint
5th International Workshop on Rapid Continuous Software Engineering and
1st International Workshop on Data-Driven Decisions, Experimentation and Evolution

In conjunction with ICSE 2019, Montreal, Canada, May 27, 2019

Deadline: Feb 1st

http://saso2019.cs.umu.se/calls/call-for-posters-and-demos/

SASO 2019
13th IEEE International Conference on Self-Adaptive and Self-Organizing Systems
June 16 - 20, 2019, Umeå, Sweden

Call for Posters and Demos

The 16th International Conference on Autonomic Computing (ICAC 2019) and the 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2019), under the federated umbrella of FAS* 2019 – Foundations and Applications of Self* Systems, in addition to their main technical programs, solicit the submission of posters and demos on specific aspects of autonomic software and

Important Dates

Workshop Proposal Deadline	**December 14, 2018**
Paper Submission Deadline	**March 10, 2019**