

# VNS and PBIG as Optimization Cores in a Cooperative Optimization Approach for Distributing Service Points

Thomas Jatschka, Tobias Rodemann, Günther Raidl

Eurocast 2019, February 20, 2019

The logo for HIRI, consisting of the letters 'H', 'I', 'R', and 'I' in a stylized, outlined font.The logo for TU WIEN, featuring the letters 'TU' above 'WIEN' in white on a blue square background.The logo for the Algorithms and Complexity Group, featuring the lowercase letters 'ac' followed by three vertical bars of increasing height.

ALGORITHMS AND  
COMPLEXITY GROUP

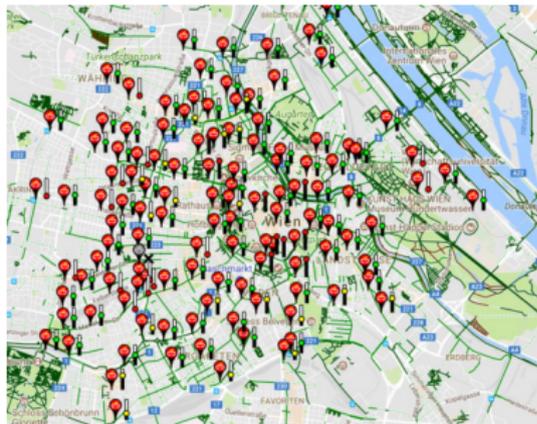
# Motivation

**Goal:** find an optimal set of locations within a certain geographical area for placing service points

# Motivation

**Goal:** find an optimal set of locations within a certain geographical area for placing service points

- ▶ for mobility purposes:
  - ▶ bike sharing stations
  - ▶ rental stations for car sharing
  - ▶ charging stations for electric vehicles
  - ▶ ...



## Related Literature

- ▶ lots of studies for related service point distribution problems:



C. Kloimüller and G. R. Raidl, “Hierarchical clustering and multilevel refinement for the bike-sharing station planning problem,” in *International Conference on Learning and Intelligent Optimization*. Springer, 2017, pp. 150–165.



I. Frade, A. Ribeiro, G. Gonçalves, and A. Antunes, “Optimal Location of Charging Stations for Electric Vehicles in a Neighborhood in Lisbon, Portugal,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2252, pp. 91–98, 2011.

# Motivation

**Where to place service points?**

# Motivation

## **Where to place service points?**

estimate demands upfront using

- ▶ demographic data,
- ▶ geographic data,
- ▶ information on public transport and the street network,
- ▶ and knowledge on diverse special locations,
- ▶ and surveys of potential customers
- ▶ ...

# Motivation

## Where to place service points?

estimate demands upfront using

- ▶ demographic data,
- ▶ geographic data,
- ▶ information on public transport and the street network,
- ▶ and knowledge on diverse special locations,
- ▶ and surveys of potential customers
- ▶ ...

⇒ **challenging and error-prone**

# Motivation

the actual usage of a service system by the user depends on

- ▶ service points on a few specific locations
- ▶ non-trivial relationships of the user's necessities and preferences in conjunction with larger parts of the whole service system

# Motivation

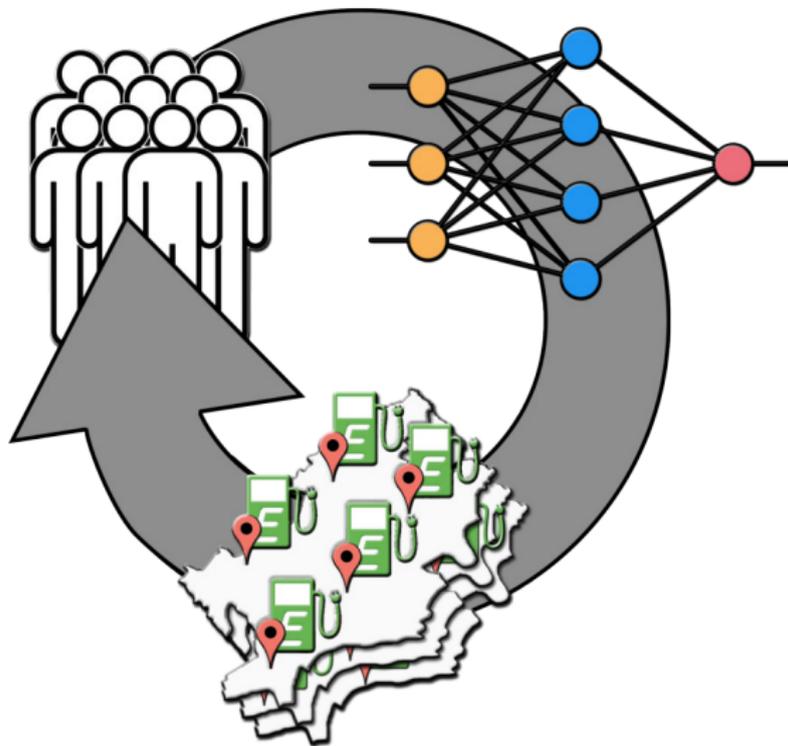
the actual usage of a service system by the user depends on

- ▶ service points on a few specific locations
- ▶ non-trivial relationships of the user's necessities and preferences in conjunction with larger parts of the whole service system

## Assumption

We are in general not able to obtain complete information from a possible user about the conditions on which how much of his potential demand will be fulfilled!

# Cooperative Optimization Approach (COA)



# Outline

## Cooperative Optimization Approach

The Service Point Distribution Problem  
Optimization Component

## Computational Results

## Conclusion and Future Work

# The Service Point Distribution Problem (SPDP)

## Problem Formalization

We are given

- ▶ a set of **locations**  $V = \{1, \dots, n\}$  at which service points may be built,
- ▶ a set of potential **users**  $U = \{1, \dots, m\}$ ,
- ▶ building **costs**  $c_v$  and maintenance costs  $z_v$  for each location  $v \in V$
- ▶ a maximum **budget**  $B$  for building service points,
- ▶ and a **prize**  $p$  that is earned for each unit of satisfied customer demand

# The Service Point Distribution Problem (SPDP)

## Objective Function

$$\max f(x) = \max \left( p \cdot \sum_{u \in U} \sum_{v \in V} d(u, v, x) - \sum_{v \in V} z_{v, x_v} \right)$$

$d(u, v, x)$  = demand of user  $u \in U$  fulfilled at service point  $v \in V$  in solution  $x$

# The Service Point Distribution Problem (SPDP)

## Objective Function

$$\max f(x) = \max \left( p \cdot \sum_{u \in U} \sum_{v \in V} d(u, v, x) - \sum_{v \in V} z_{v, x_v} \right)$$

$d(u, v, x)$  = demand of user  $u \in U$  fulfilled at service point  $v \in V$  in solution  $x$

⇒ can only be determined by directly asking the user

# The Service Point Distribution Problem (SPDP)

## Surrogate Function

- ▶ it is not reasonable for a user to evaluate every generated solution
- ▶ it is in practice not possible to explicitly express  $d(u, v, x)$

# The Service Point Distribution Problem (SPDP)

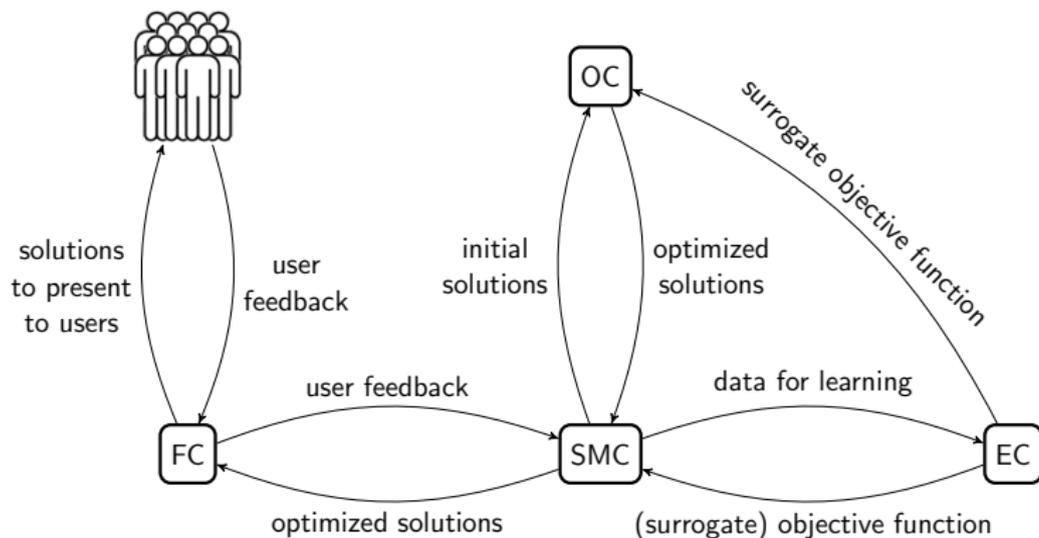
## Surrogate Function

- ▶ it is not reasonable for a user to evaluate every generated solution
- ▶ it is in practice not possible to explicitly express  $d(u, v, x)$

⇒ surrogate function  $\tilde{d}(u, v, x)$

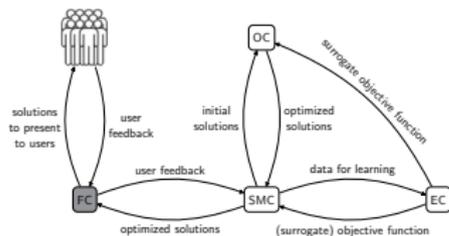
# Cooperative Optimization Approach (COA)

## Framework



# Cooperative Optimization Approach (COA)

## Framework

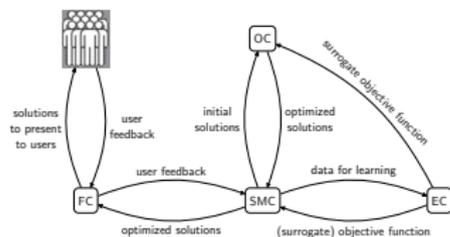


## Feedback Component:

- ▶ selects for each user solutions to evaluate
- ▶ solutions derived from best solutions in SMC
- ▶ discover new relevant locations for a user
- ▶ discover relationship between relevant locations

# Cooperative Optimization Approach (COA)

## Framework

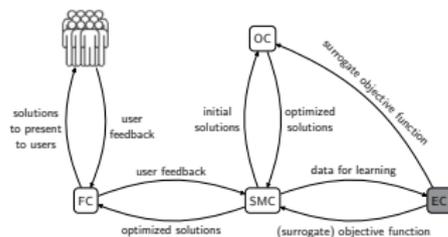


## Users:

- ▶ evaluate solutions
- ▶ users specify how often they intend to use the service points in the presented solution
- ▶  $d(u, v, x)$

# Cooperative Optimization Approach (COA)

## Framework



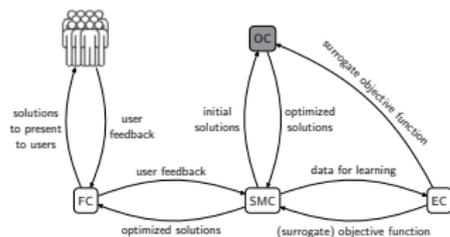
## Evaluation Component<sup>1</sup>:

- ▶ generates a surrogate function  $\tilde{d}(u, v, x)$  from user feedback
- ▶ surrogate function consists of multiple machine learning models  $g_{u,v}(x)$
- ▶ iteratively improved

<sup>1</sup>to appear in proceedings of EvoCOP 2019

# Cooperative Optimization Approach (COA)

## Framework

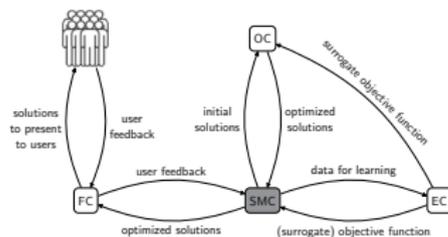


## Optimization Component:

- ▶ generates new/optimized solutions
- ▶ using the surrogate function to evaluate solution

# Cooperative Optimization Approach (COA)

## Framework



## Solution Management Component:

- ▶ stores solutions, user feedback, (surrogate) objective values, . . .
- ▶ provides mechanisms for detecting duplicate solutions

# Optimization Component

- ▶ black-box optimization model
- ▶ uses surrogate objective function for evaluating solutions
- ▶ goal: yield one or more close-to-optimal solutions w.r.t. surrogate function

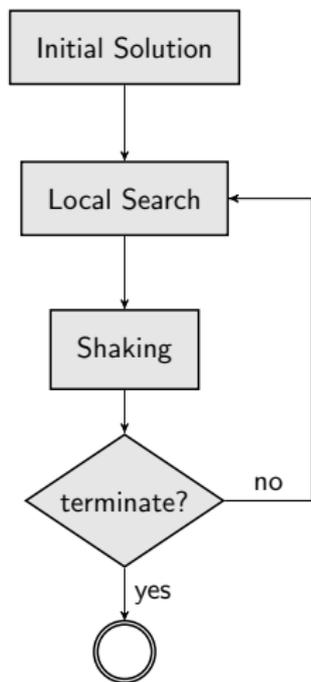
# Optimization Component

- ▶ black-box optimization model
- ▶ uses surrogate objective function for evaluating solutions
- ▶ goal: yield one or more close-to-optimal solutions w.r.t. surrogate function

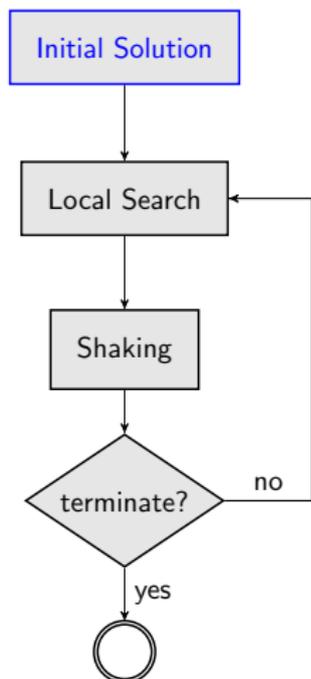
## **Implemented Algorithms:**

- ▶ Variable Neighborhood Search (VNS)
- ▶ Population Based Iterated Greedy Algorithm (PBIG)

# Variable Neighborhood Search (VNS)



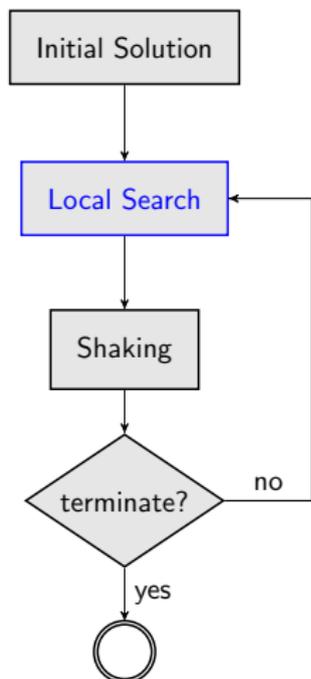
# Variable Neighborhood Search (VNS)



## Initial Solution:

- ▶ randomized construction heuristic,
- ▶ or solution from SMC

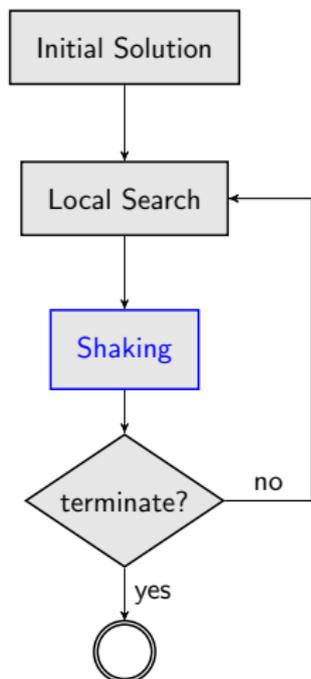
# Variable Neighborhood Search (VNS)



## Local Search:

- ▶ only feasible solutions are considered
- ▶ first improvement
- ▶ exchange & complete neighborhood:
  1. replace a location in the solution with an unused location
  2. add randomly chosen locations as long as the budget allows it

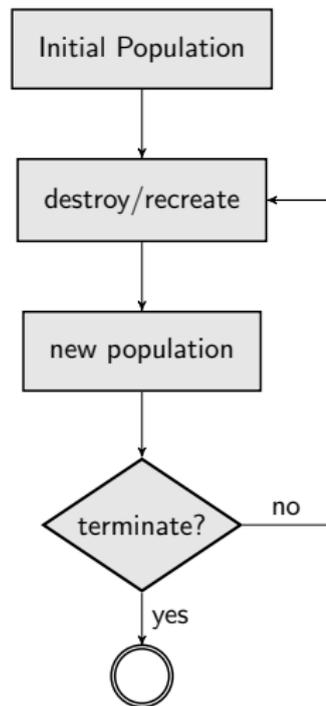
# Variable Neighborhood Search (VNS)



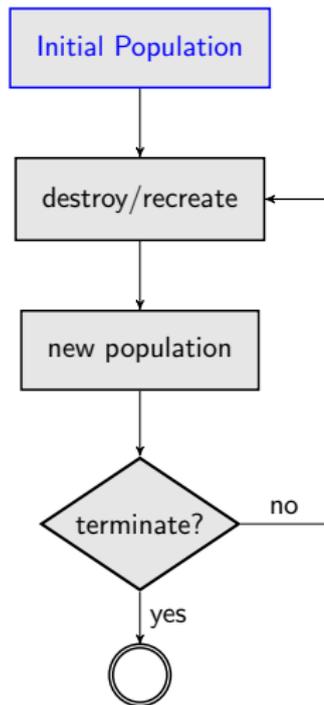
## Shaking:

- ▶ shaking neighborhood  $k$ :
  1. remove  $k$  stations uniformly at random
  2. reinsert stations in a random order until no more stations can be added

# Population Based Iterated Greedy Algorithm (PBIG)



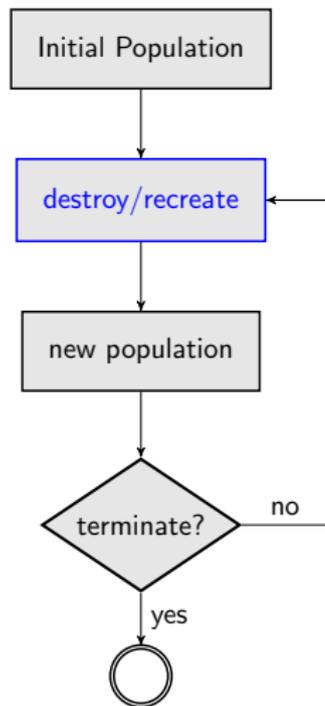
# Population Based Iterated Greedy Algorithm (PBIG)



## Initial Population:

- ▶ randomized construction heuristic,
- ▶ or solutions from SMC

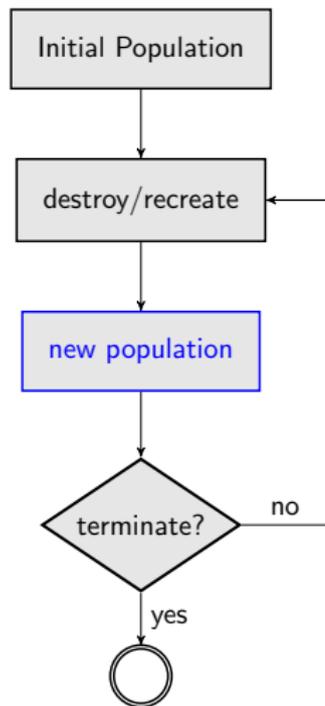
# Population Based Iterated Greedy Algorithm (PBIG)



## destroy/recreate:

- ▶ destroy/recreate operations:
  - ▶ exchange & complete neighborhood
  - ▶ shaking neighborhoods

# Population Based Iterated Greedy Algorithm (PBIG)



## new population:

- ▶ keep solutions with the highest objectives

# VNS vs. PBIG

## VNS:

- ▶ returns a single strongly optimized solution
- ▶ low number of solutions in SMC
- ▶ low diversification in solutions in FC and EC

## PBIG:

- ▶ returns multiple (possibly less) optimized solutions
- ▶ high number of solutions in SMC
- ▶ high diversification in solutions in FC and EC

# Test Scenario

- ▶ proof-of-concept implementation with simulated (perfectly reasonable) users
  - ▶ inspired distributing charging stations for EVs
  - ▶ each user has a set of **use case locations**
  - ▶ nearby service points can satisfy demand of use case locations
  - ▶ satisfied demand depends on distance to use case location
- ⇒ **can be solved exactly with mixed integer programming**

# Computational Experiments

- ▶ Programming Language: C++, Python
- ▶ Test runs have been executed on an Intel Xeon E5-2640 v4 with 2.40GHz
- ▶ 11 sets of test instances with different number of locations  $n$  and number of users  $m$
- ▶ 330 benchmark instances in total

# VNS vs. PBIG

## Parameters

# VNS vs. PBIG

## Parameters

### VNS:

- ▶ initial solution: randomized construction heuristic
- ▶ termination criterion: 60 iterations without improvement
- ▶ shaking neighborhoods:  
 $k \in \{1, 2\}$

### PBIG:

- ▶ initial solution: randomized construction heuristic
- ▶ population size: 100
- ▶ termination criterion: 300 iterations/3 generations without improvement
- ▶ shaking neighborhoods:  
 $k \in \{1, \dots, 10\}$

# VNS vs. PBIG

## Results

$n$	$m$	VNS				PBIG			
		$\overline{\%}\text{-gap}$	$\sigma_{\%}\text{-gap}$	$\overline{n}_{it}^{\text{best}}$	$t[s]$	$\overline{\%}\text{-gap}$	$\sigma_{\%}\text{-gap}$	$\overline{n}_{it}^{\text{best}}$	$t[s]$
50	50	<b>0.26</b>	0.48	<b>29</b>	<b>3</b>	0.28	1.22	696	8
50	60	<b>0.20</b>	0.61	<b>35</b>	<b>4</b>	0.31	1.36	656	9
50	70	<b>0.00</b>	0.02	<b>32</b>	<b>4</b>	0.10	0.42	634	8
50	80	0.31	0.74	<b>31</b>	<b>4</b>	<b>0.09</b>	0.28	631	9
50	90	<b>0.14</b>	0.42	<b>32</b>	<b>4</b>	0.37	1.01	648	11
50	100	0.37	1.03	<b>33</b>	<b>4</b>	<b>0.01</b>	0.07	706	15
60	50	0.25	0.64	<b>43</b>	<b>4</b>	<b>0.07</b>	0.34	862	9
70	50	0.34	0.62	<b>54</b>	<b>5</b>	<b>0.28</b>	0.57	1113	17
80	50	0.43	0.54	<b>56</b>	<b>6</b>	<b>0.24</b>	0.51	1389	23
90	50	0.30	0.42	<b>64</b>	<b>7</b>	<b>0.19</b>	0.60	1628	30
100	50	<b>0.37</b>	0.47	<b>65</b>	<b>9</b>	0.42	0.82	1754	39

# COA[VNS] vs. COA[PBIG]

## Parameters

### VNS:

- ▶ randomized construction heuristic
- ▶ termination criterion: 60 iterations without improvement
- ▶ shaking neighborhoods:  
 $k \in \{1, 2\}$

### PBIG:

- ▶ population size: 100
- ▶ 100 initial solutions from SMC
- ▶ termination criterion: 300 iterations/3 generations without improvement
- ▶ shaking neighborhoods:  
 $k \in \{1, \dots, 10\}$

### COA:

- ▶ termination criterion: 5 iterations without improvement or after 7200s

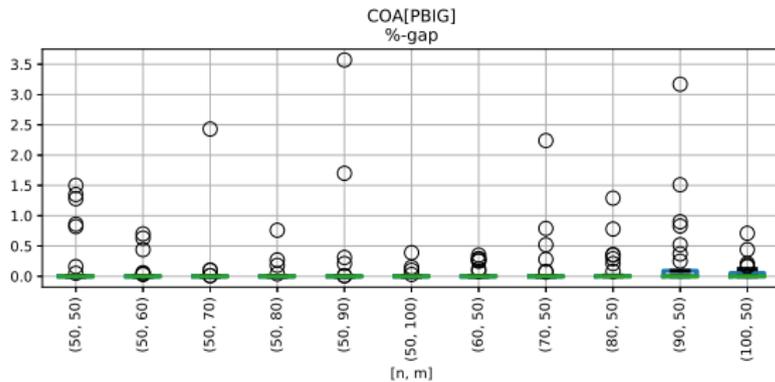
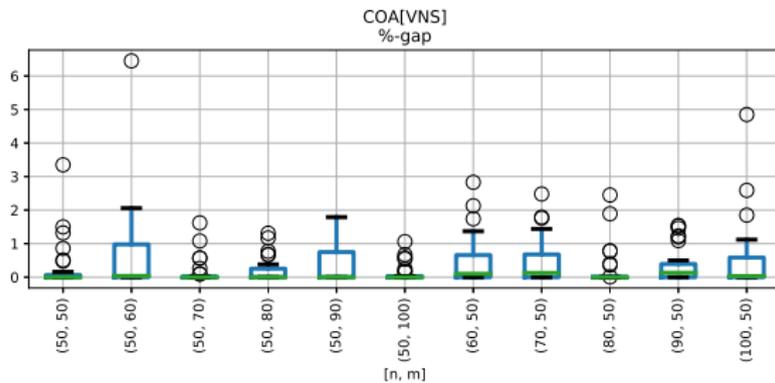
# COA[VNS] vs. COA[PBIG]

## Results

$n$	$m$	COA[VNS]				COA[PBIG]			
		$\overline{\%}\text{-gap}$	$\sigma\% \text{-gap}$	$\overline{n}_{it}$	$t[s]$	$\overline{\%}\text{-gap}$	$\sigma\% \text{-gap}$	$\overline{n}_{it}$	$t[s]$
50	50	0.28	0.70	11	<b>2259</b>	<b>0.20</b>	0.45	<b>10</b>	2662
50	60	0.73	1.27	<b>9</b>	<b>2343</b>	<b>0.06</b>	0.18	<b>9</b>	2643
50	70	0.14	0.37	<b>10</b>	<b>3107</b>	<b>0.09</b>	0.44	<b>10</b>	3764
50	80	0.19	0.36	<b>10</b>	<b>3588</b>	<b>0.04</b>	0.15	<b>10</b>	3919
50	90	0.42	0.68	10	<b>3596</b>	<b>0.19</b>	0.71	<b>9</b>	4516
50	100	0.12	0.26	<b>10</b>	<b>4391</b>	<b>0.02</b>	0.08	<b>10</b>	4995
60	50	0.48	0.72	11	<b>2460</b>	<b>0.05</b>	0.11	<b>10</b>	2944
70	50	0.46	0.66	11	<b>2533</b>	<b>0.13</b>	0.43	<b>10</b>	3658
80	50	0.22	0.58	12	<b>2864</b>	<b>0.11</b>	0.28	<b>11</b>	4810
90	50	0.37	0.52	<b>11</b>	<b>2910</b>	<b>0.26</b>	0.65	<b>11</b>	5435
100	50	0.49	1.02	12	<b>3460</b>	<b>0.07</b>	0.15	<b>11</b>	7197

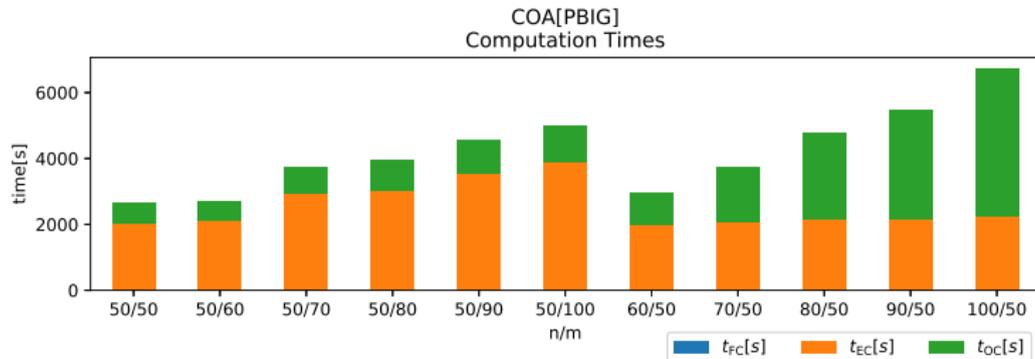
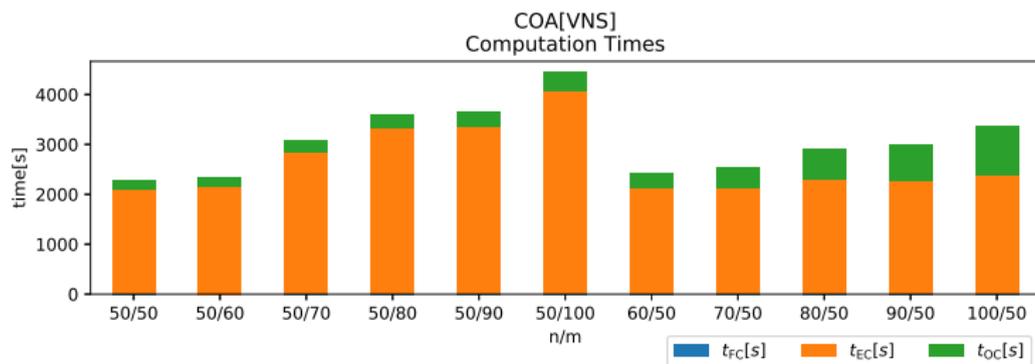
# COA[VNS] vs. COA[PBIG]

optimality gaps



# COA[VNS] vs. COA[PBIG]

Computation times



# Conclusion and Future Work

## Conclusion

- ▶ proof-of-concept implementation for a cooperative optimization algorithm (COA) for the SPDP
- ▶ tested VNS and PBIG as core optimization of COA
- ▶ VNS as well as PBIG generate close to optimal solutions
- ▶ COA[PBIG] generates better results than COA[VNS] but does not scale very well

## Future Work

- ▶ determine limits of COA w.r.t. scalability
- ▶ different optimization algorithms
- ▶ white/gray box model

Thank you for your attention!  
Questions?

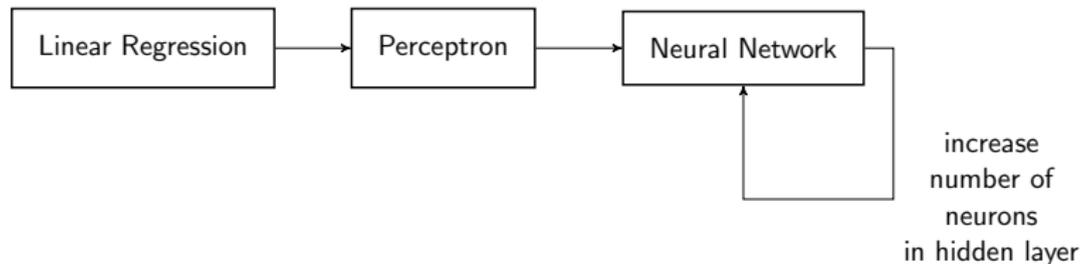
# Evaluation Component

Solution:

$$x = (x_v)_{v \in V}$$

Surrogate function:

$$\tilde{d}(u, v, x) = \begin{cases} 0 & \text{if } x_v = 0 \vee v \notin V_u \\ \max(0, g_{u,v}(x)) & \text{else} \end{cases}$$



# COA[VNS] vs. COA[PBIG]

Results - tciter 300

$n$	$m$	COA[VNS]				COA[PBIG]			
		$\overline{\%}\text{-gap}$	$\sigma_{\%}\text{-gap}$	$\overline{n}_{it}$	$t[s]$	$\overline{\%}\text{-gap}$	$\sigma_{\%}\text{-gap}$	$\overline{n}_{it}$	$t[s]$
50	50	0.76	1.49	<b>10</b>	<b>2427</b>	<b>0.20</b>	0.45	<b>10</b>	2662
50	60	0.36	0.75	10	2930	<b>0.06</b>	0.18	<b>9</b>	<b>2643</b>
50	70	0.29	0.48	<b>10</b>	<b>3648</b>	<b>0.09</b>	0.44	<b>10</b>	3764
50	80	0.30	0.60	<b>10</b>	3984	<b>0.04</b>	0.15	<b>10</b>	<b>3919</b>
50	90	0.56	0.90	10	4733	<b>0.19</b>	0.71	<b>9</b>	<b>4516</b>
50	100	0.19	0.48	<b>10</b>	5107	<b>0.02</b>	0.08	<b>10</b>	<b>4995</b>
60	50	0.42	0.72	11	3212	<b>0.05</b>	0.11	<b>10</b>	<b>2944</b>
70	50	0.26	0.51	11	<b>3519</b>	<b>0.13</b>	0.43	<b>10</b>	3658
80	50	0.44	0.92	12	4836	<b>0.11</b>	0.28	<b>11</b>	<b>4810</b>
90	50	0.36	0.71	<b>11</b>	5475	<b>0.26</b>	0.65	<b>11</b>	<b>5435</b>
100	50	0.37	0.89	<b>10</b>	<b>7106</b>	<b>0.07</b>	0.15	11	7197