A Cooperative Optimization Approach for Distributing Service Points in Mobility Applications

Thomas Jatschka, Tobias Rodemann, Günther Raidl

EvoCOP 2019, April 25, 2019

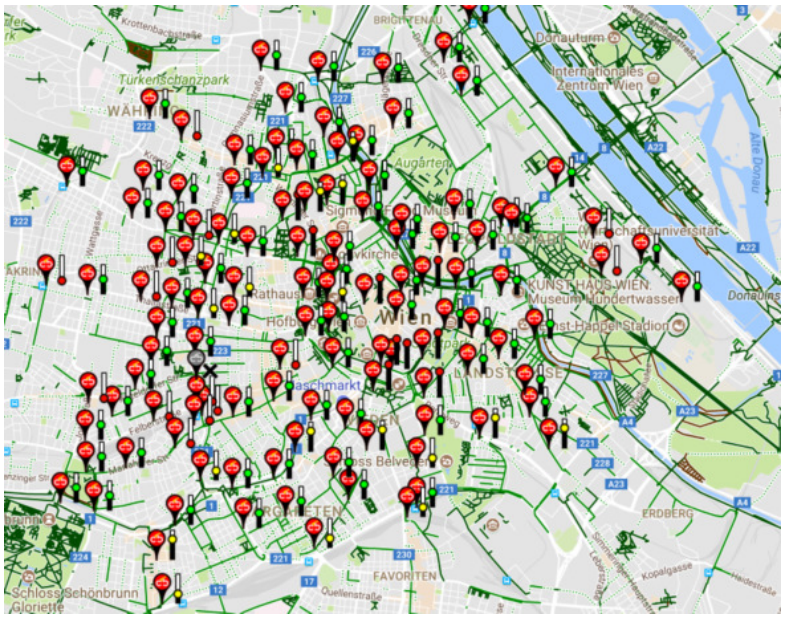
Motivation

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. Kloimüllner 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
- ...

$\Rightarrow$ 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

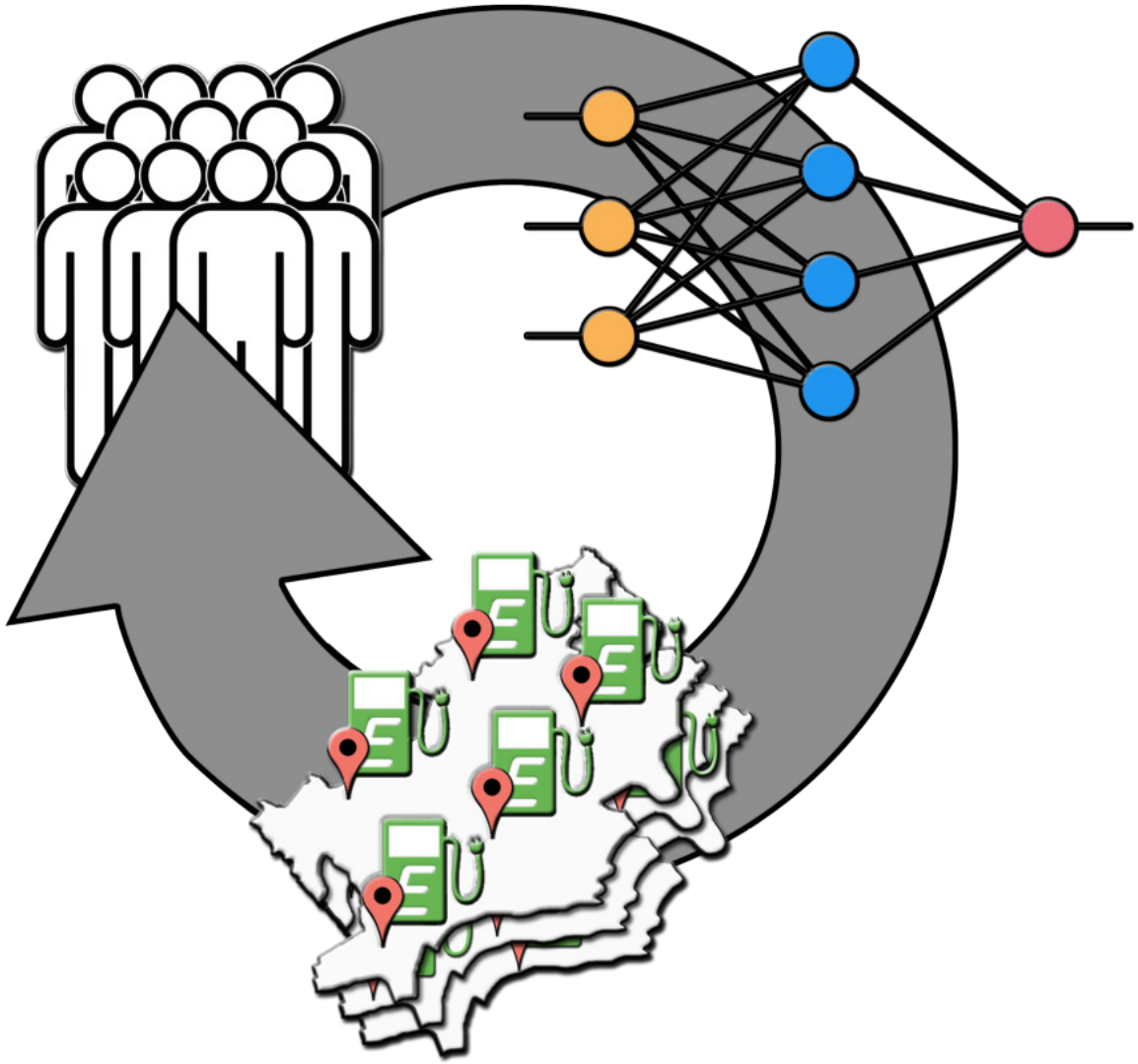
the actual usage of a service system by the user depends on

- 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

The Service Point Distribution Problem

Cooperative Optimization Approach
- Feedback Component
- Evaluation Component
- Optimization Component

Computational Results

Conclusion and Future Work

The Service Point Distribution Problem (SPDP)

Problem Formalization

We are given

- a set of locations $V = \{1, \ldots, n\}$ at which service points may be built,
- a set of potential users $U = \{1, \ldots, m\}$,
- 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

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

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

The Service Point Distribution Problem (SPDP)

Objective Function

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

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

$\Rightarrow$ can only be determined by directly asking the user

The Service Point Distribution Problem (SPDP)

Surrogate Function

- it is not reasonable of 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)

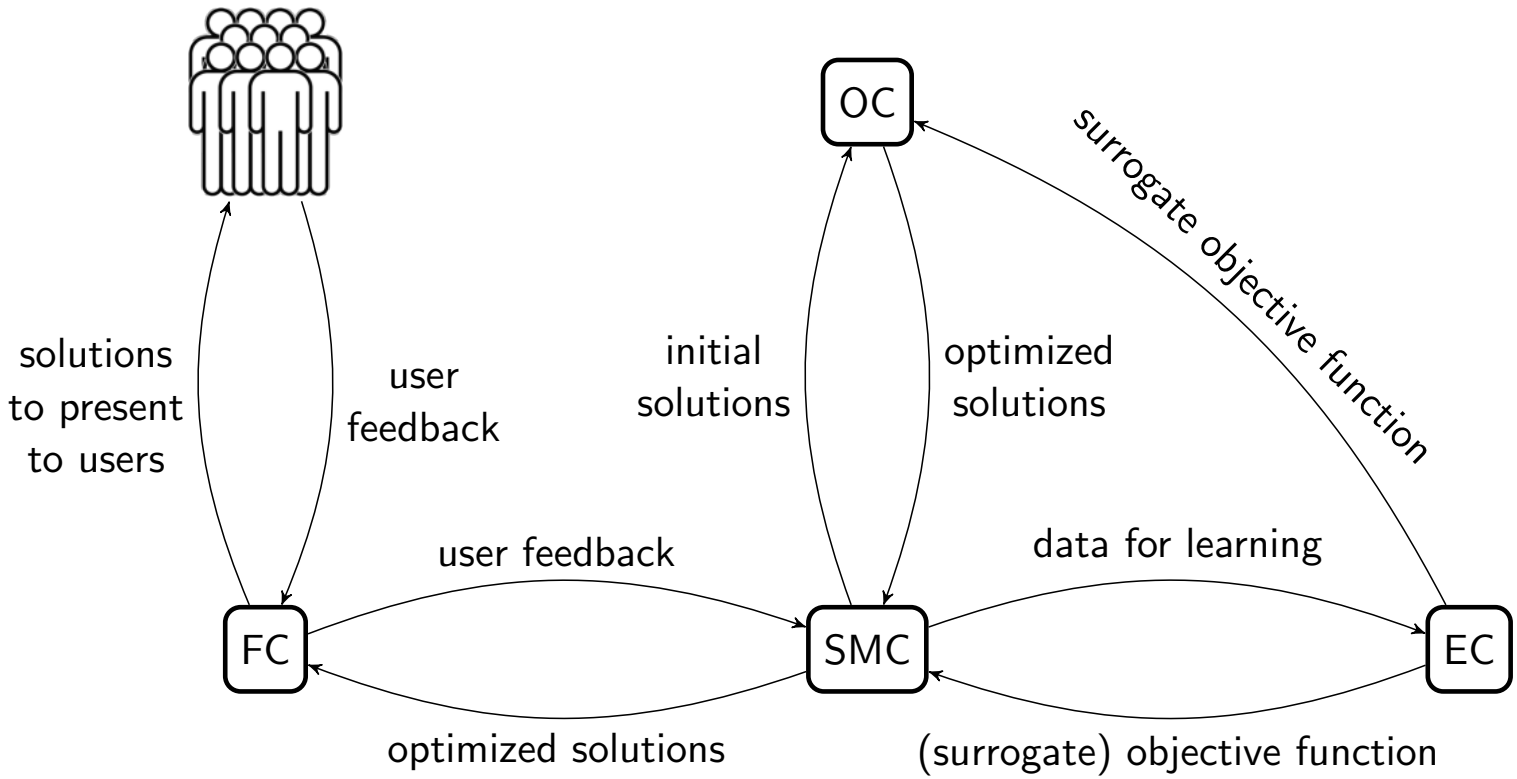
Surrogate Function

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

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

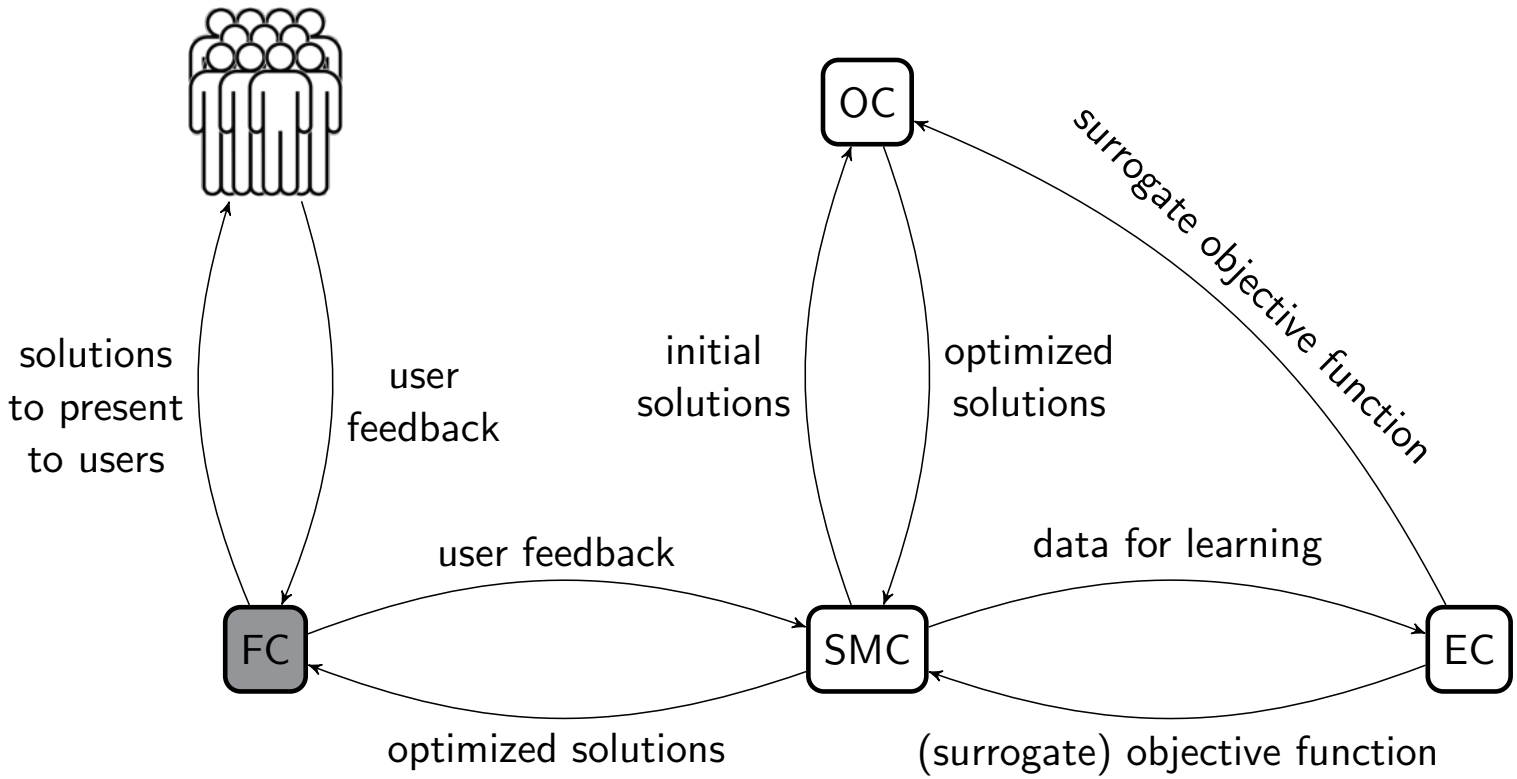
Cooperative Optimization Approach (COA)

Framework



Cooperative Optimization Approach (COA)

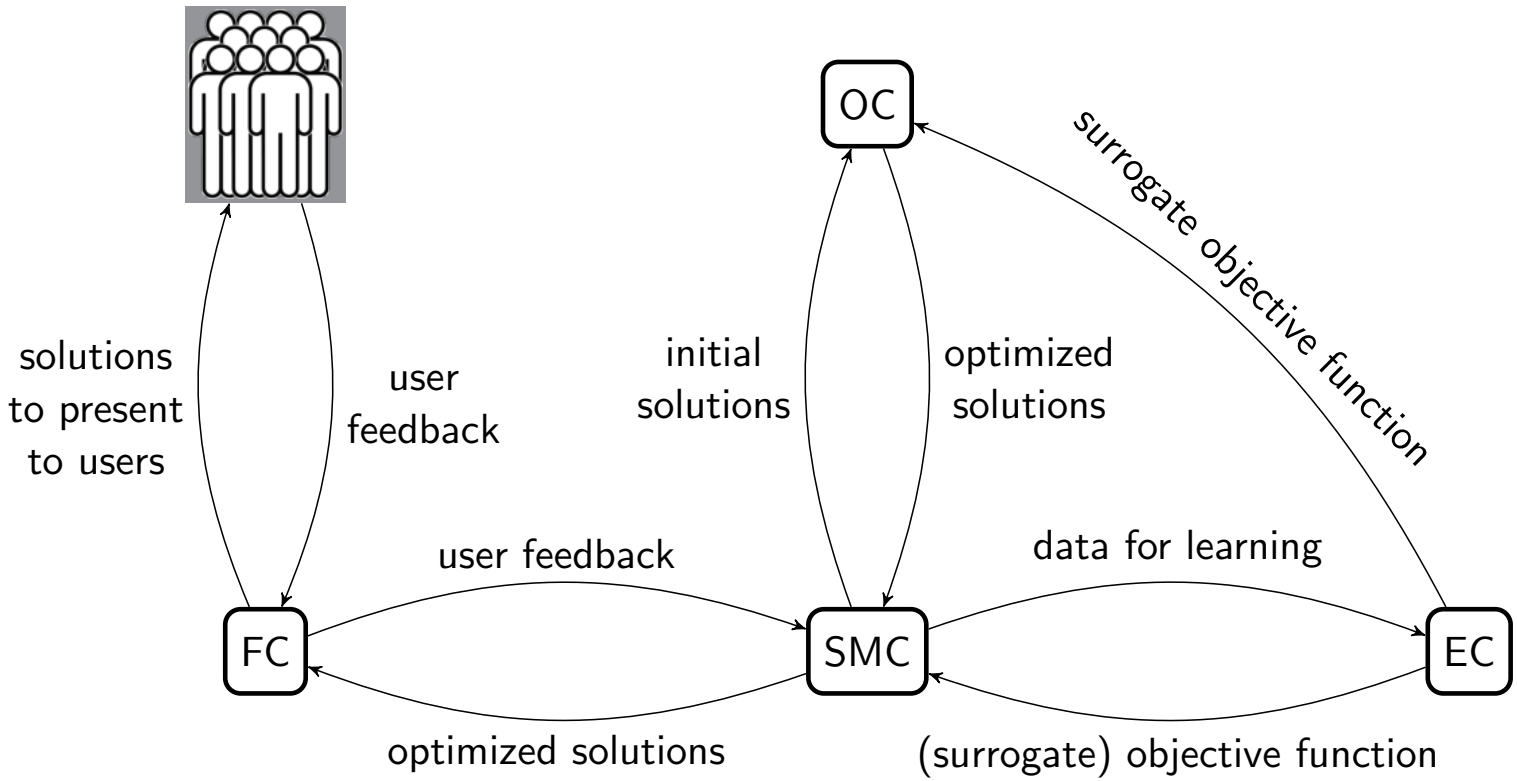
Framework



Feedback Component: selects for each user solutions to evaluate

Cooperative Optimization Approach (COA)

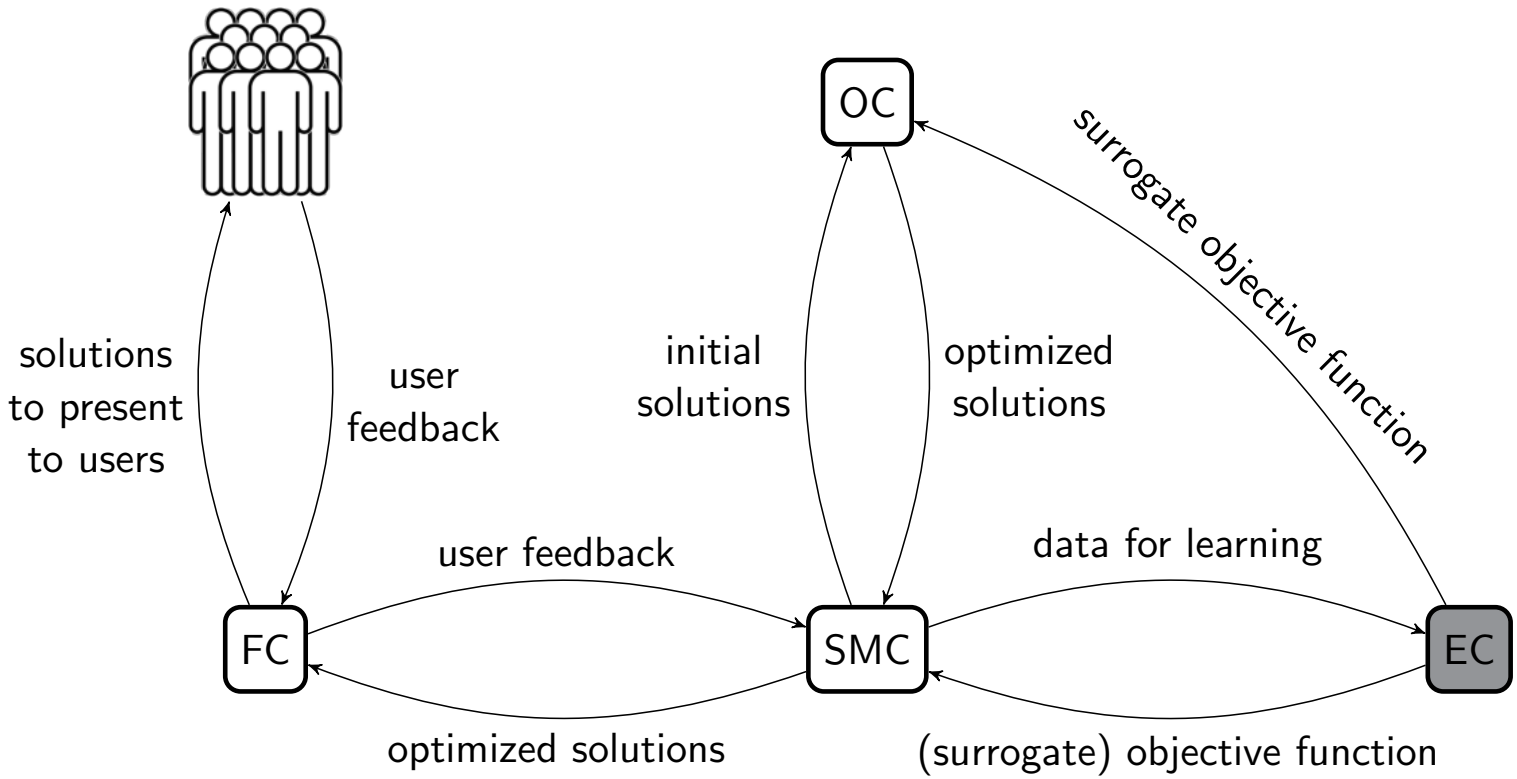
Framework



Users: user simulation, evaluates solutions

Cooperative Optimization Approach (COA)

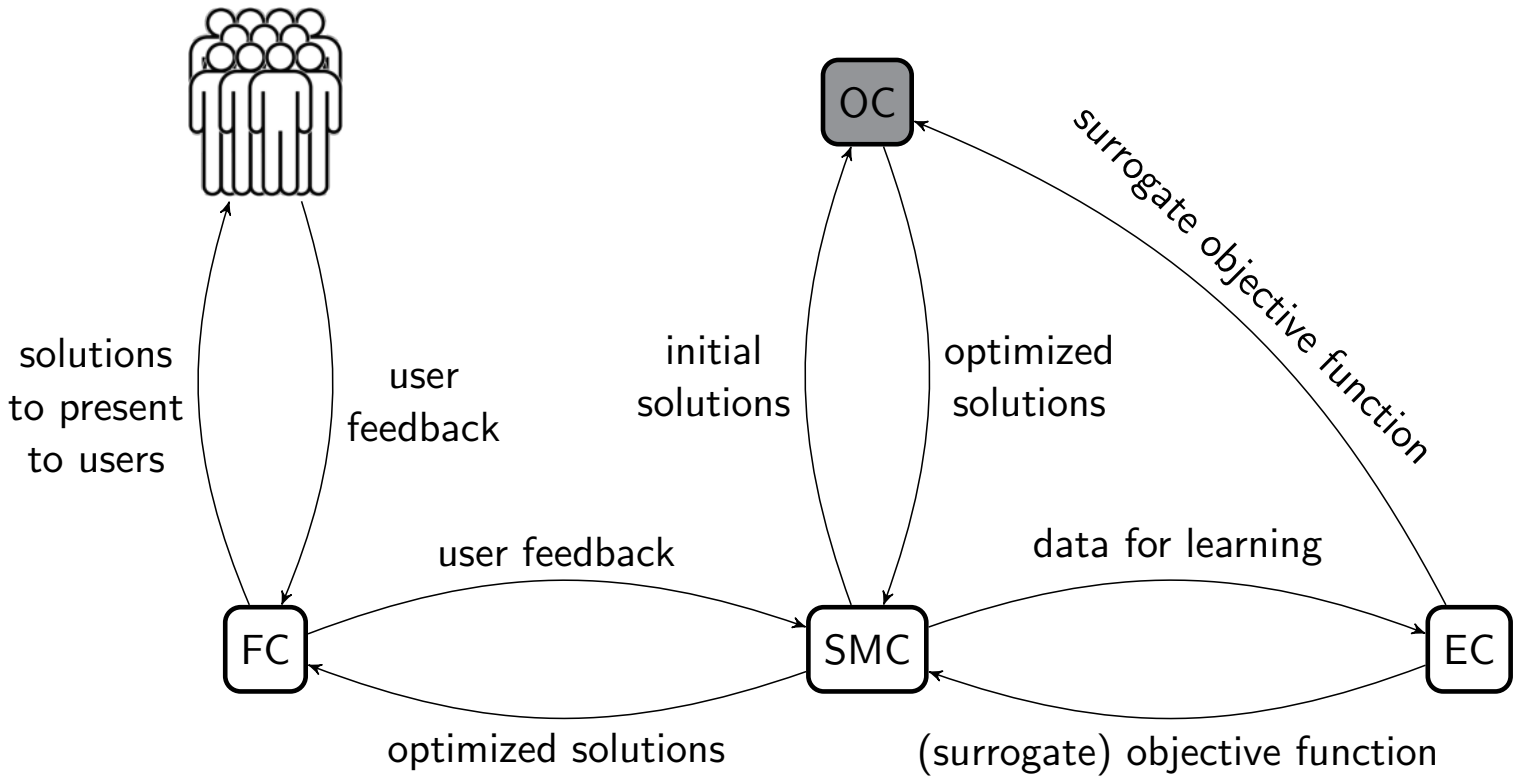
Framework



Evaluation Component: generates a surrogate function from user feedback

Cooperative Optimization Approach (COA)

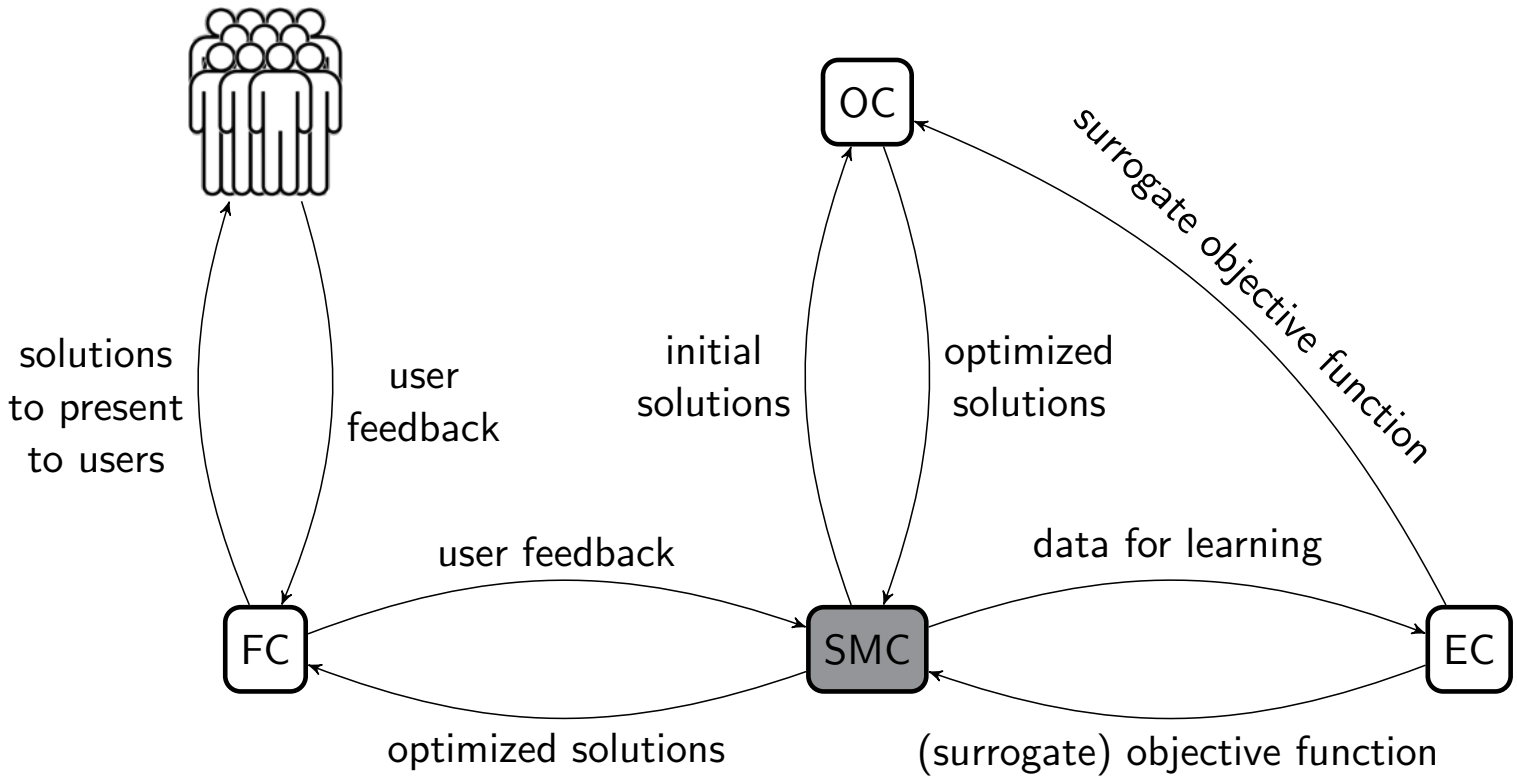
Framework



Optimization Component: generates solutions using the surrogate function to evaluate solutions

Cooperative Optimization Approach (COA)

Framework



Solution Management Component: stores solutions and user feedback

Feedback Component

- selects for each user solutions to evaluate

Feedback Component

- selects for each user solutions to evaluate

Goals:

- identify relevant locations for a user $u : V_u$
- gain information on relationship between relevant locations
- get exact evaluations for most promising solutions

Feedback Component

Strategies for generating solutions

Strategies for generating solutions:

- Best Solution Strategy: choose the γ best solutions
- Irrelevant Locations Strategy: select all locations for which user has never indicated any positive demand
- Best Solution Mutation Strategy:
  1. get best solution x
  2. remove all relevant locations V_u of user u from x
  3. add new random locations to x

Evaluation Component

- 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

Evaluation Component

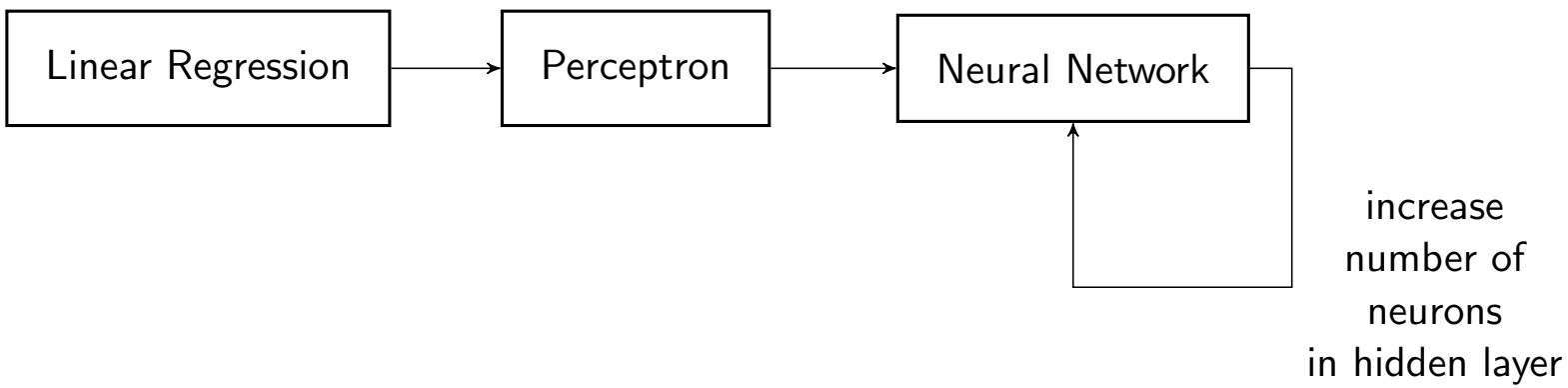
Solution:

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

Surrogate function:

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

$g_{u,v}$:

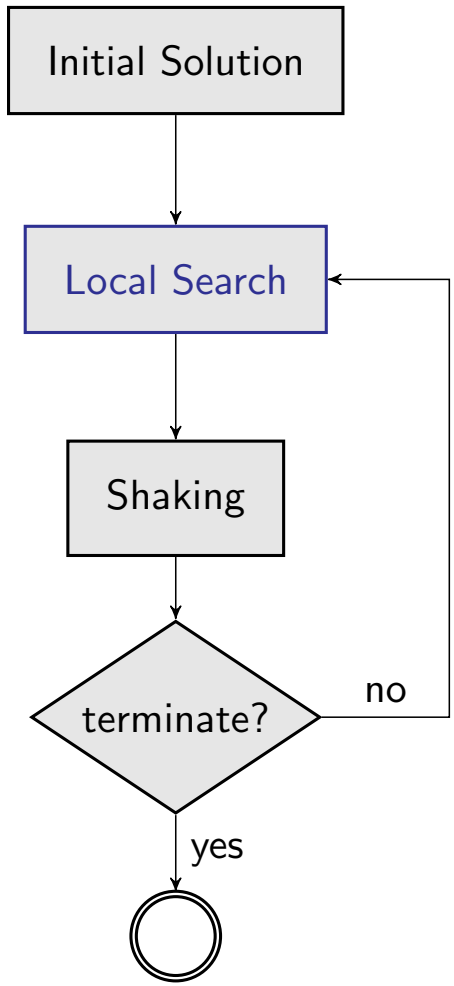


Optimization Component

- 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
- Variable Neighborhood Search (VNS)

Optimization Component

Variable Neighborhood Search (VNS)

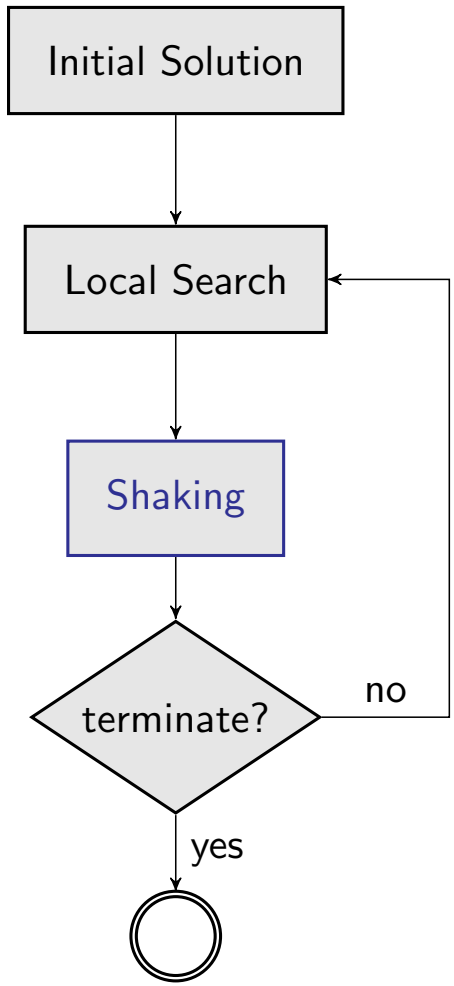


Local Search:

1. replace a location in the solution with an unused location
2. add randomly chosen locations as long as the budget allows it

Optimization Component

Variable Neighborhood Search (VNS)



Local Search:

1. replace a location in the solution with an unused location
2. add randomly chosen locations as long as the budget allows it

Shaking neighborhood k:

1. remove k random service points
2. insert random service points until no more service points can be added

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

$\Rightarrow$ **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

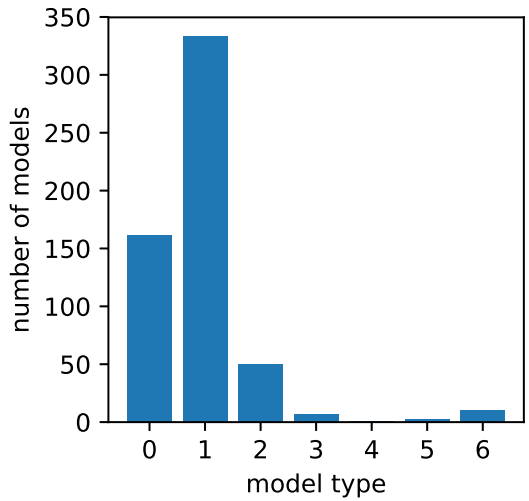
Results

n	m	$\overline{n_{\text{it}}}$	$\overline{\%\text{-gap}}$	$\sigma_{\%\text{-gap}}$	$\overline{\%\text{-}\Delta\tilde{f}}$	$\sigma_{\%\text{-}\Delta\tilde{f}}$	time[s]
50	50	10	0.34	0.67	0.19	0.14	2063
50	60	10	0.38	1.05	0.31	0.19	2594
50	70	11	0.21	0.58	0.33	0.34	2936
50	80	10	0.28	0.51	0.35	0.32	3522
50	90	10	0.56	0.89	0.37	0.34	3867
50	100	11	0.28	0.48	0.41	0.44	4424
60	50	11	0.46	1.43	0.53	1.35	2335
70	50	11	0.49	0.93	0.38	0.48	2447
80	50	12	0.49	0.73	0.27	0.28	2904
90	50	11	0.27	0.40	0.23	0.19	2946
100	50	13	0.26	0.44	0.28	0.27	3889

Results

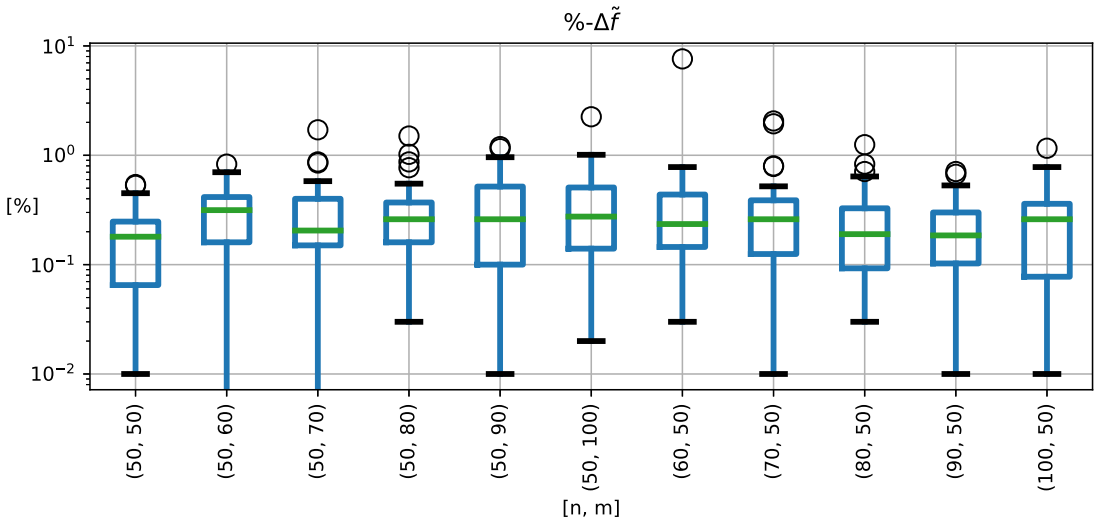
Surrogate function in the final iteration

Distribution of the machine learning models



0. . . linear regression model
1. . . perceptron
2-6. . . neural network

Percentage errors to best found solution



Conclusion and Future Work

Conclusion

- proof-of-concept implementation for a cooperative optimization algorithm (COA) for the SPDP
- machine learning models are able to reliably learn non-trivial user behavior
- optimization is able to find solutions with only small remaining optimality gaps

Conclusion and Future Work

Conclusion

- proof-of-concept implementation for a cooperative optimization algorithm (COA) for the SPDP
- machine learning models are able to reliably learn non-trivial user behavior
- optimization is able to find solutions with only small remaining optimality gaps

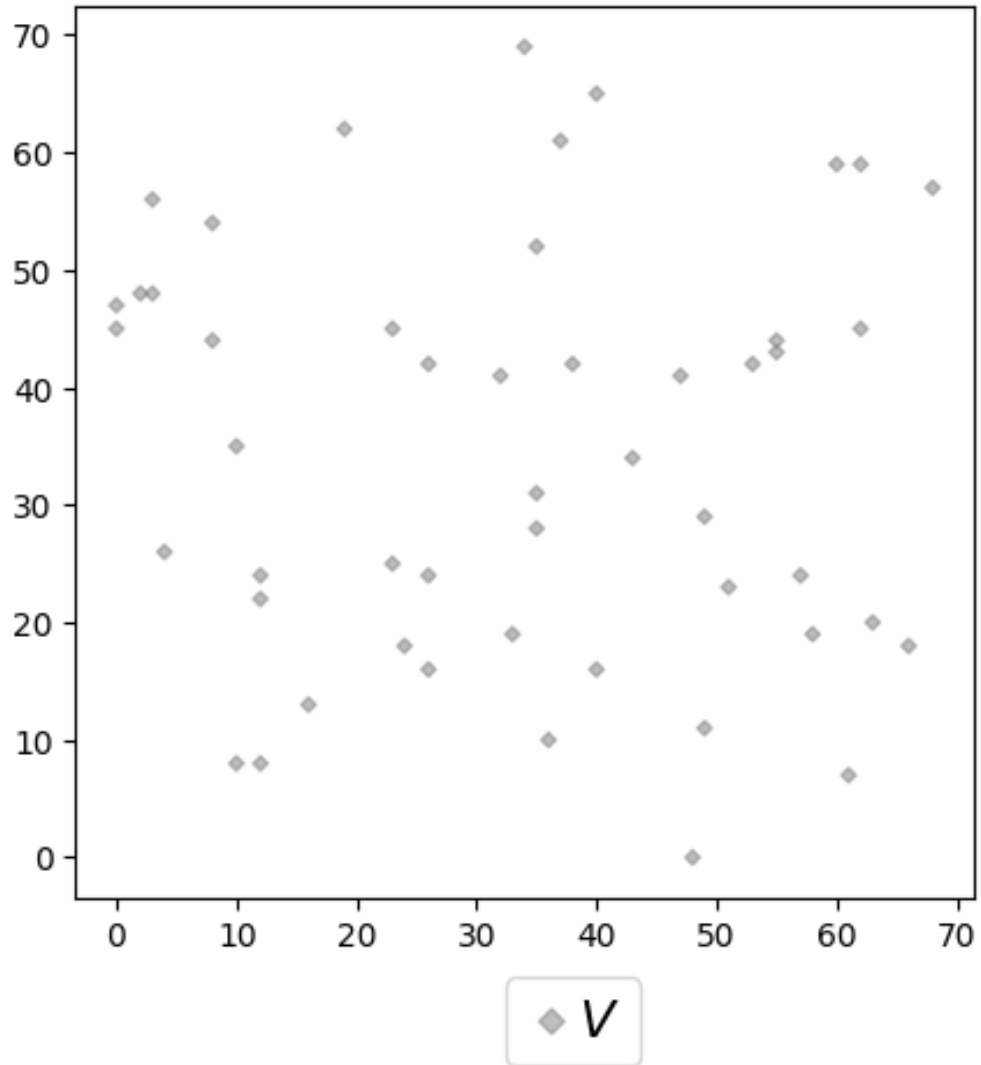
Future Work

- improve scalability of COA
- reduce the number of machine learning models in the surrogate function
- reduce the number of user evaluations
- consider different application scenarios, such as bike sharing
- consider different SPDP variants, e.g. capacitated SPDP

Thank you for your attention!
Questions?

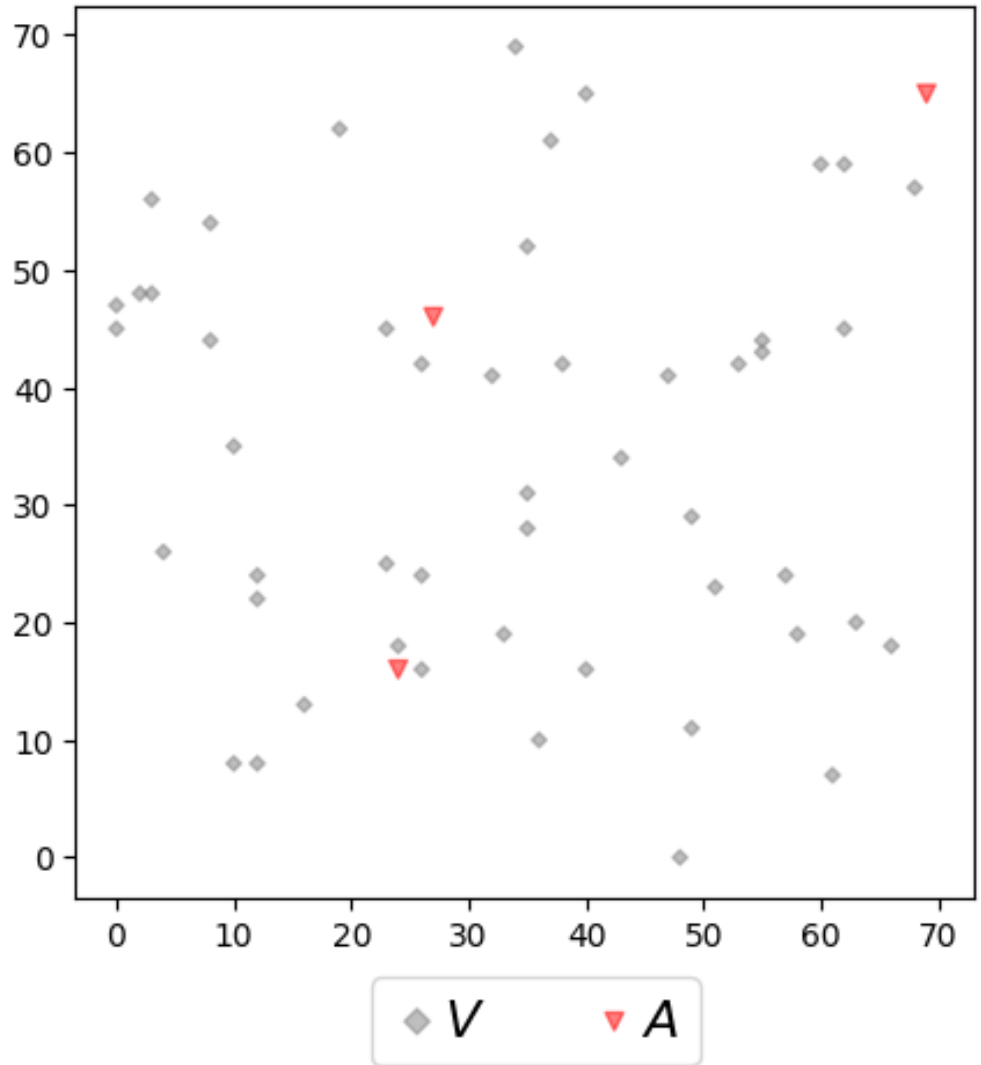
Test Scenario

- randomly generate set of potential service point locations V



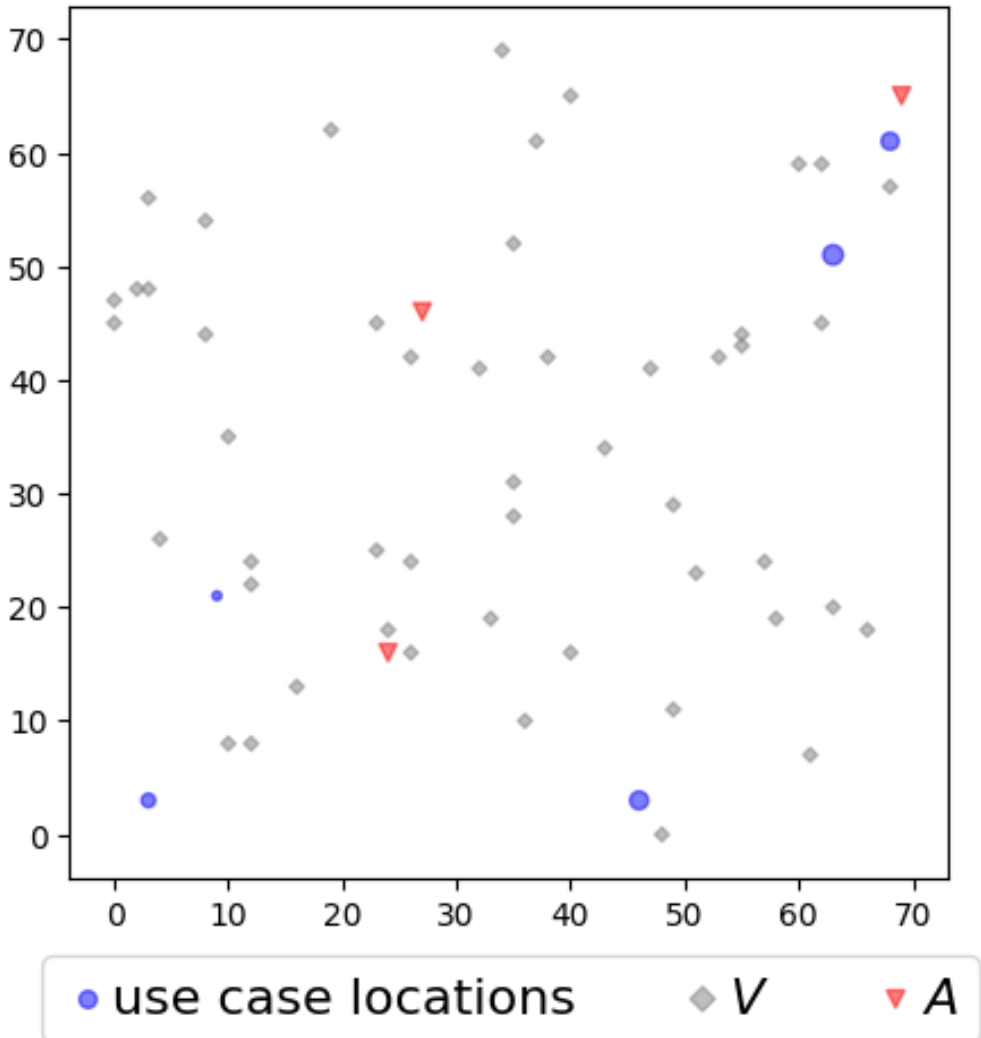
Test Scenario

- randomly generate attraction points



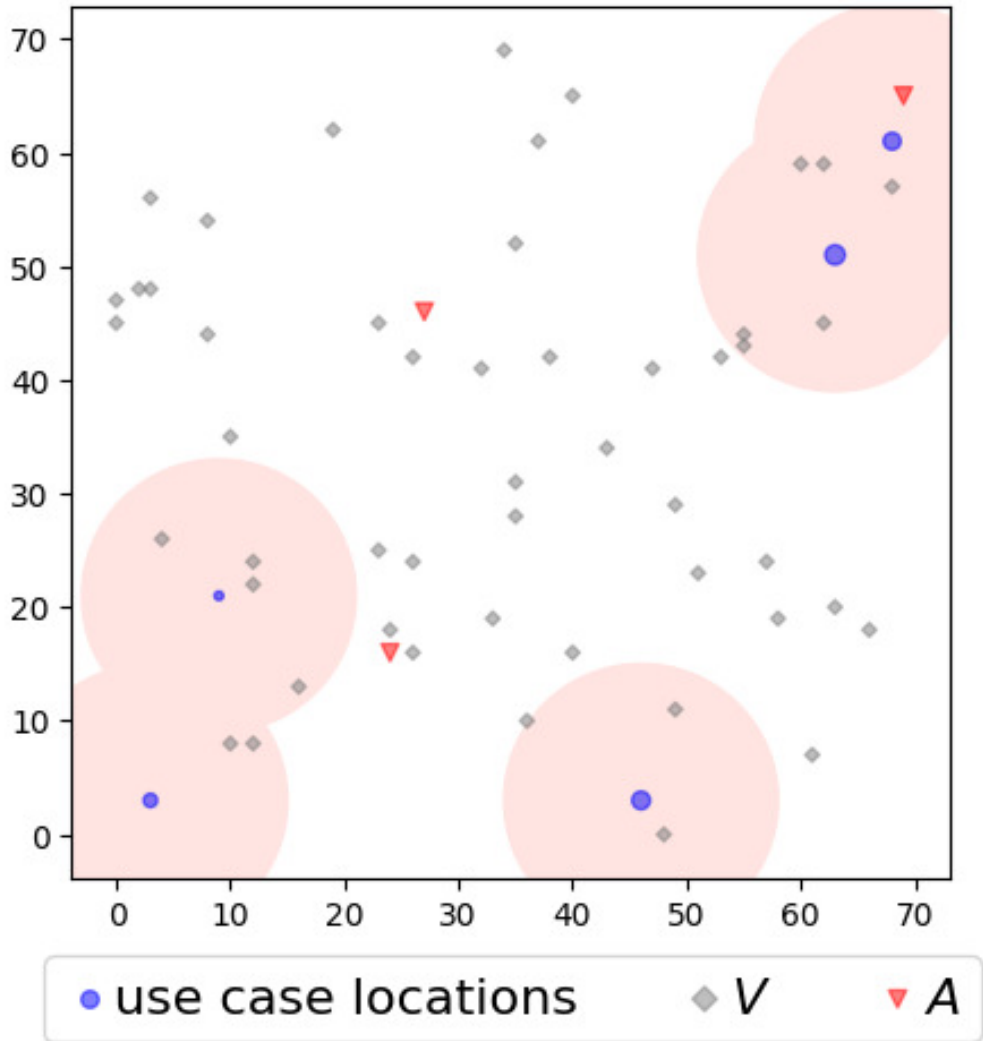
Test Scenario

- generate use case locations



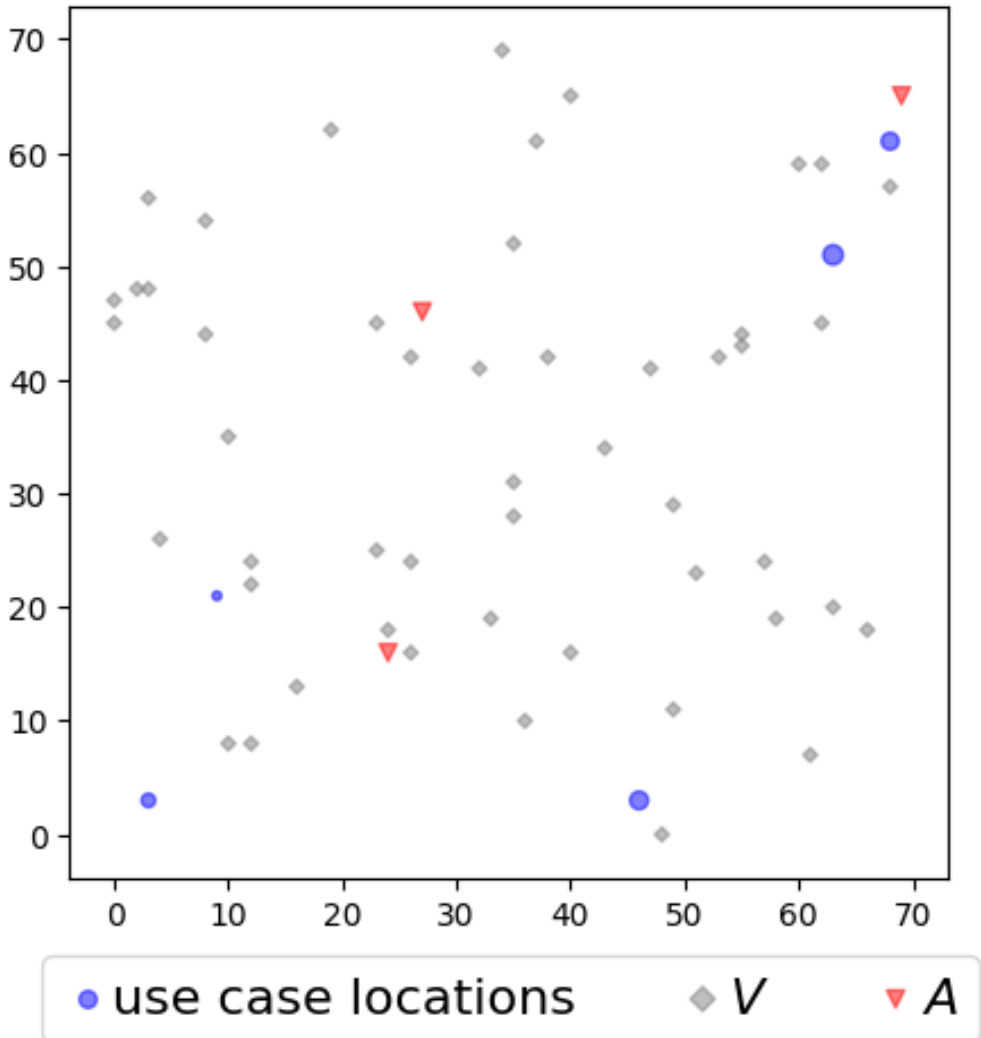
Test Scenario

- maximum walking distance of a user



Test Scenario

- generate use case locations



Test Scenario

- generate use case locations for all users