Technische Universität Braunschweig

Inventory Routing for Bike Sharing Systems

mobil.TUM 2016 – Transforming Urban Mobility

Technische Universität München, June 6-7, 2016

Jan Brinkmann, Marlin W. Ulmer, Dirk C. Mattfeld

Agenda

- Motivation
- Problem Definition
- Two-dimensional Decomposition Approach
  - Temporal Dimension
  - Spatial Dimension
- Case Studies
- Summary and Outlook

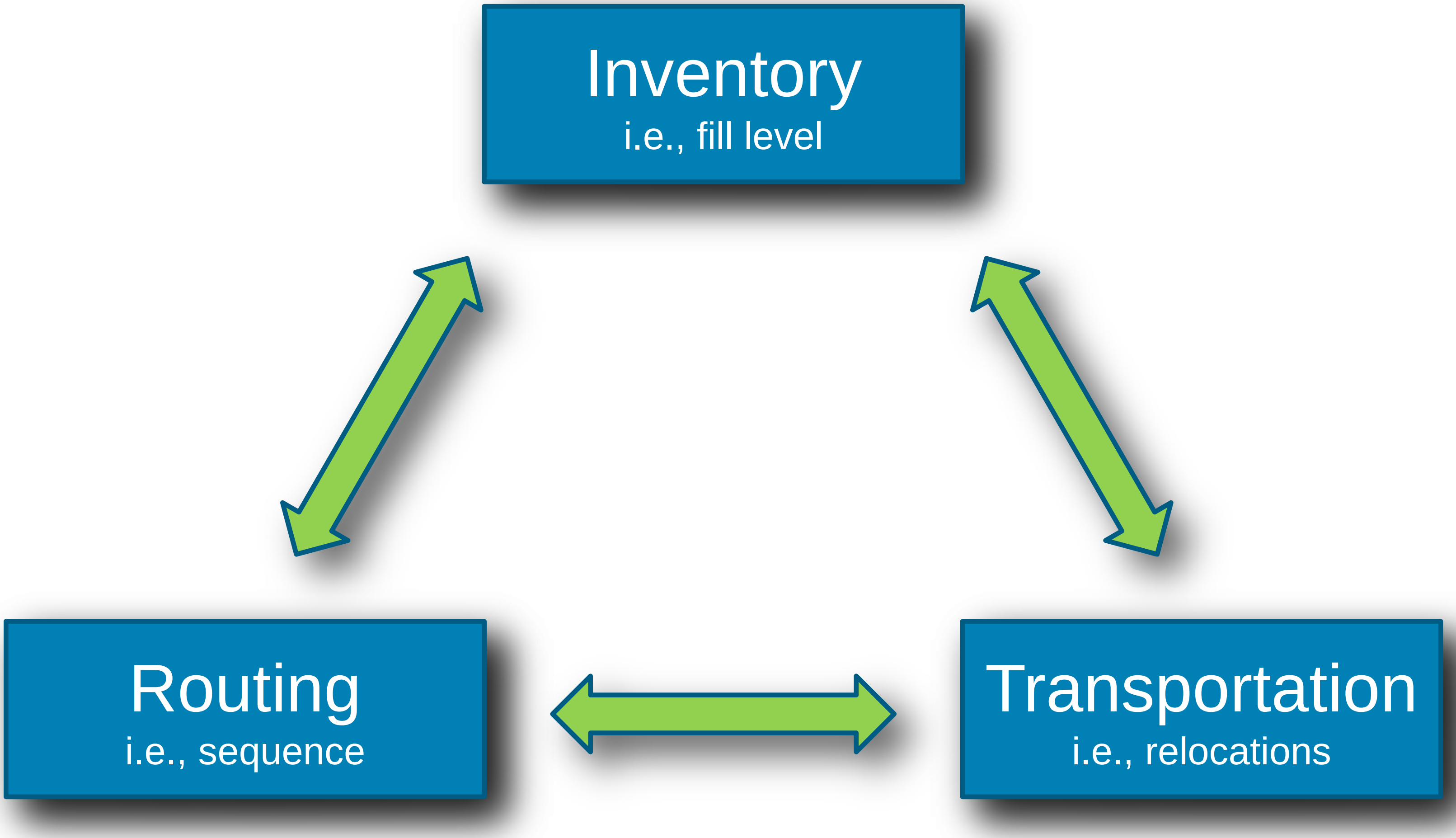
Motivation: Bike-Sharing Systems

- Public bike rental
- Short usage time
- One-way trips
- Trips, i.e.,
  - Rental request
  - Return request
- **Spatio-temporal variation of requests**

Motivation

- **Problem**
  - Discrepancy of rental and return requests lead stations either to congest or to run out of bikes.
  - Rental requests fail at empty stations.
  - Return requests fail at full stations.
- **Provider‘s view**
  - Needs to satisfy as many requests as possible.
  - Relocates bikes via transport vehicles.
  - Draws on target intervals provided by external information systems.
- **Challenges**
  - Interdependent delivery amounts, due to balancing contraints.
  - Interdependent replenishment times, due to routing.

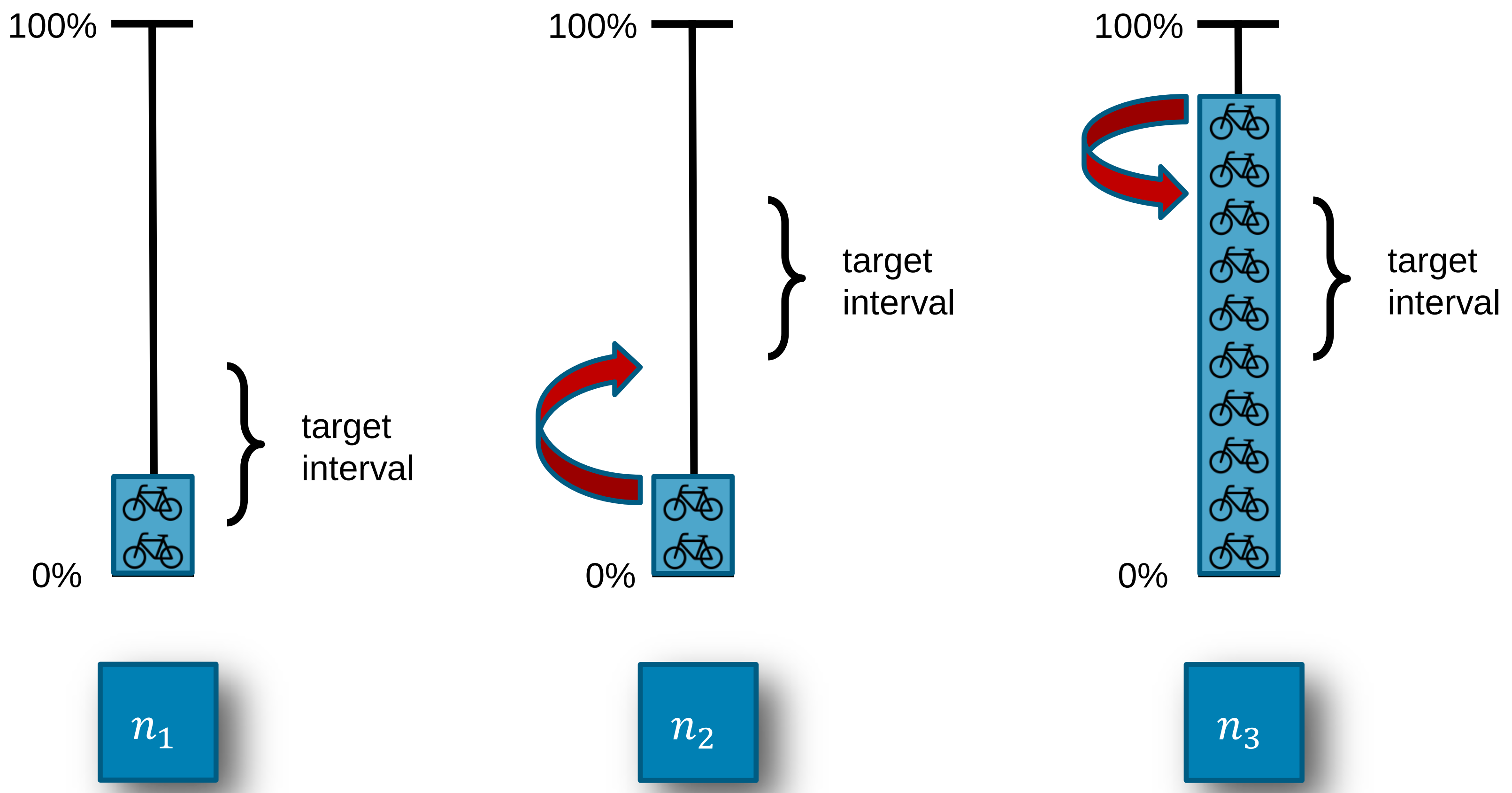
Problem Definition: Inventory Routing



Problem Definition: Sets and Functions

- Bike Sharing System
  - Set of stations: $N = \{n_0, \dots, n_{max}\}$
  - Capacity: $r: N \rightarrow \mathbb{N}_0$
  - Initial fill levels: $\mathrm{f}: N \rightarrow \mathbb{N}_0$
  - Distances: $d: N \times N \rightarrow \mathbb{R}^+$
  - Bikes: $B = \{b_0, \dots, b_{max}\}$
  - Planning horizon: $T = \{t_0, \dots, t_{max}\}$
- Expected user activities
  - Rental: $R^- = \{r_0^-, \dots, r_{max}^-\}$ $r^- = (t, n)$
  - Return: $R^+ = \{r_0^+, \dots, r_{max}^+\}$ $r^+ = (t, n)$
- Target Intervals
  - Upper Limits $\overline{\tau}: N \times T \rightarrow \mathbb{N}_0$
  - Lower Limits $\underline{\tau}: N \times T \rightarrow \mathbb{N}_0$
- Optimization
  - Set of vehicles: $V = \{v_o, \dots, v_{max}\}$
  - Capacity: $c: V \rightarrow \mathbb{N}$
  - Relocation operations
    - Pickups: $P = \{p_0, \dots, p_{max}\}$ $p = (h, n, b)$
    - Deliveries: $D = \{d_0, \dots, d_{max}\}$ $d = (h, n, b)$

Problem Definition: Fill Levels and Target Intervals



In the presence of large gaps, we assume a high probability of unsatisfied requests.

Objective: Minimize the squared gaps over all stations.

Two-dimensional Decomposition Approach

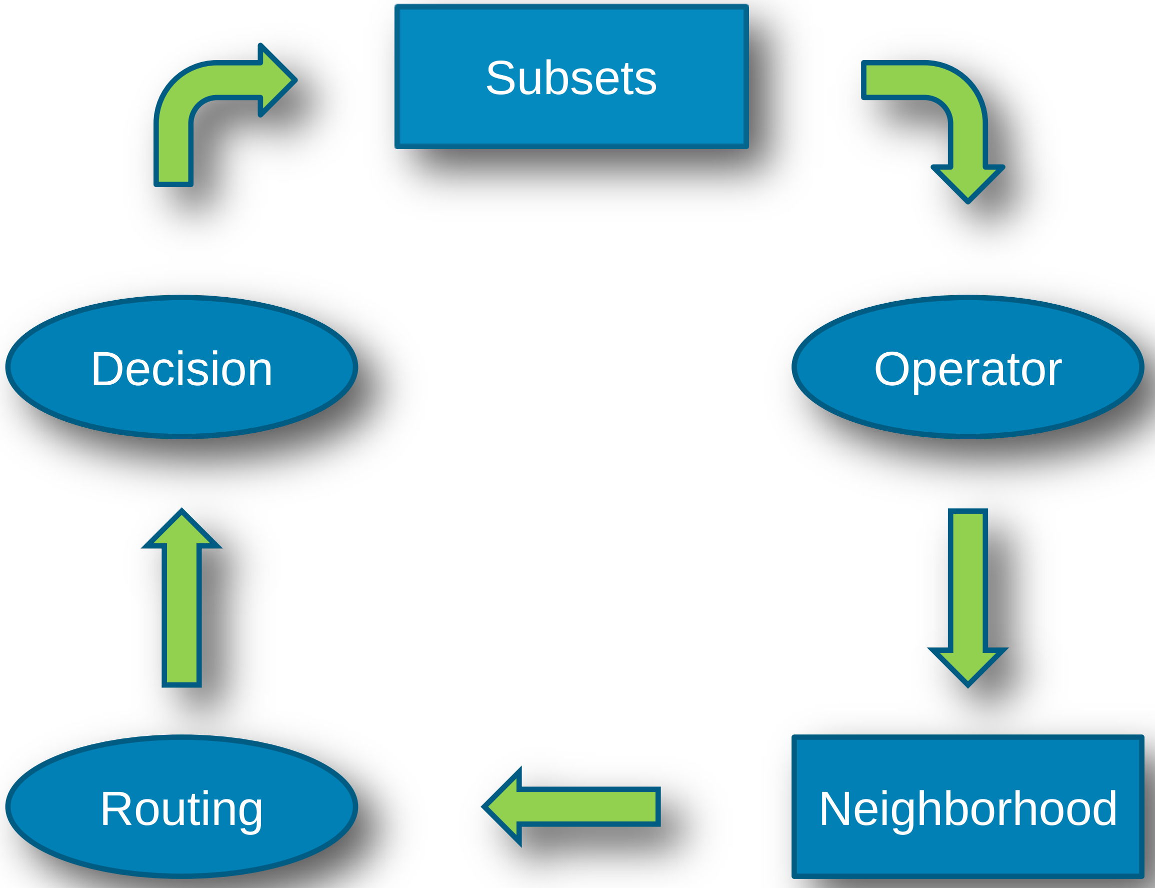
Divide the IRP into several subproblems.

- **Temporal dimension**
  - Divide planning horizon into periods.
  - Solve periods sequentially
- **Spatial dimension**
  - Divide set of stations into subsets
  - Assign each subset to one vehicle
  - For each vehicle / subset, determine a tour and relocation operations

➢ **Challenge**: Find proper subsets allowing efficient rebalancing.

Spatial Decomposition: Set Partitioning

Generate proper subsets via iterative local search proceedure:



Spatial Decomposition: Operators

Operator

Operators span a neighborhood around a current solution.

- **Insert**
  - Removes one station from it‘s subset.
  - Inserts these station in an other subset.
- ⇒ Small neighborhood
- ⇒ Can change subsets‘ sizes

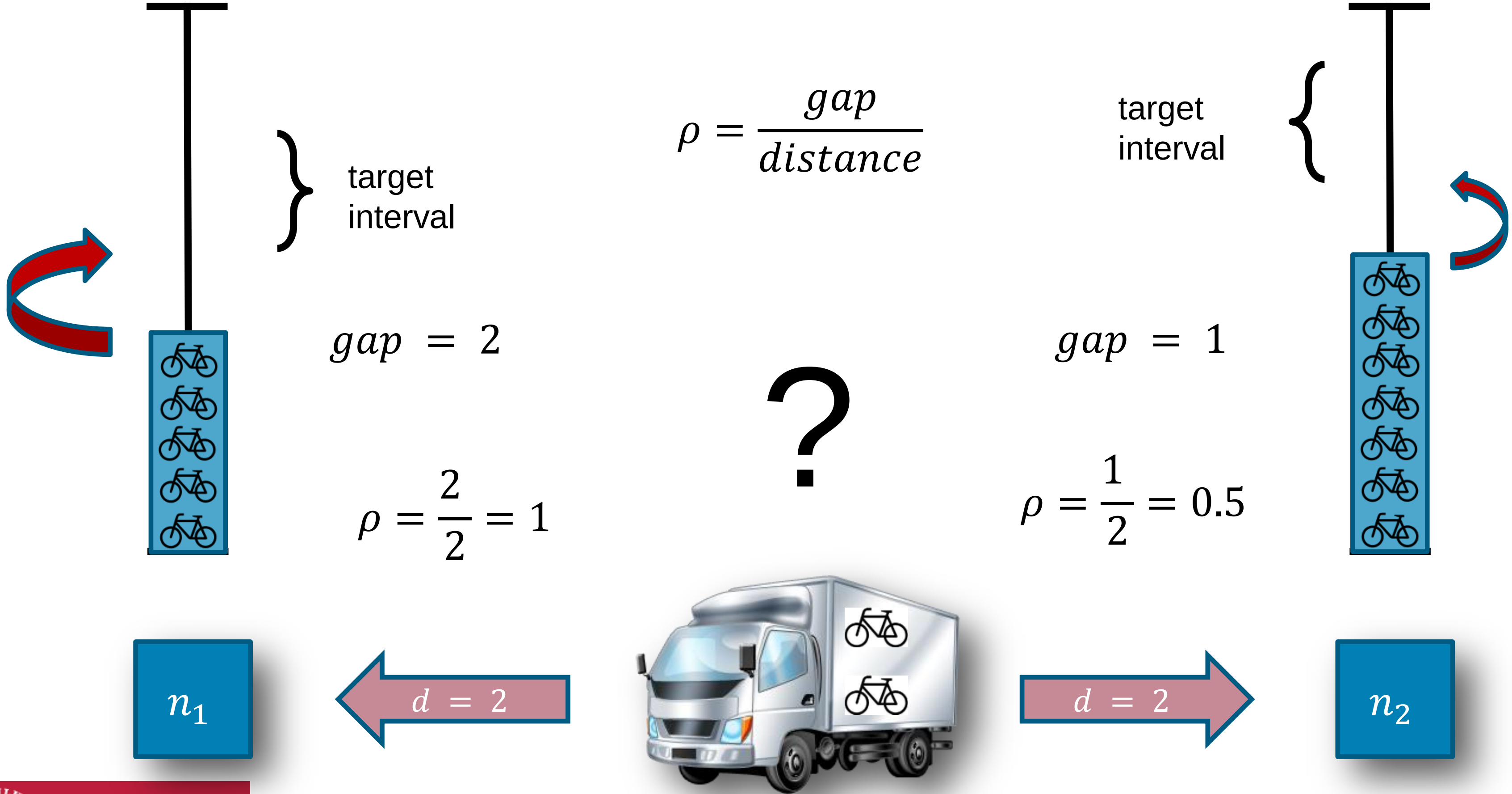
- **Exchange**
  - Removes two stations from their subsets.
  - Exchanges station‘s assignments.
- ⇒ Large neighborhood
- ⇒ Cannot change subsets‘ sizes

Spatial Decomposition: Routing

Routing

Routing evaluates subsets.

Adapted Nearest Neighbor:

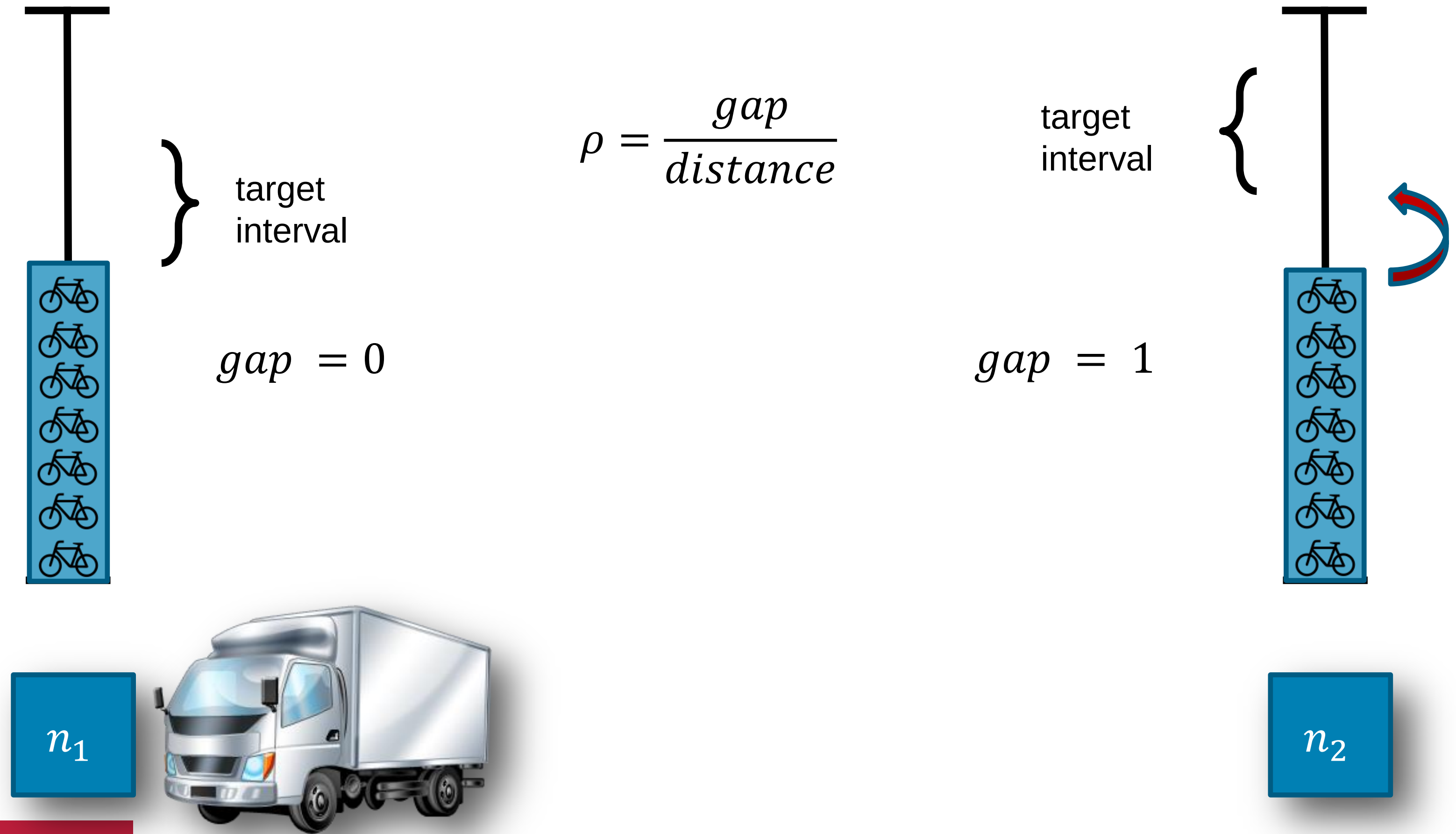


Spatial Decomposition: Routing

Routing evaluates subsets.

Adapted Nearest Neighbor:

$$\rho = \frac{gap}{distance}$$



Spatial Decomposition: Decison Making

Decision

Choosing new solutions from the current solutions neighborhood.

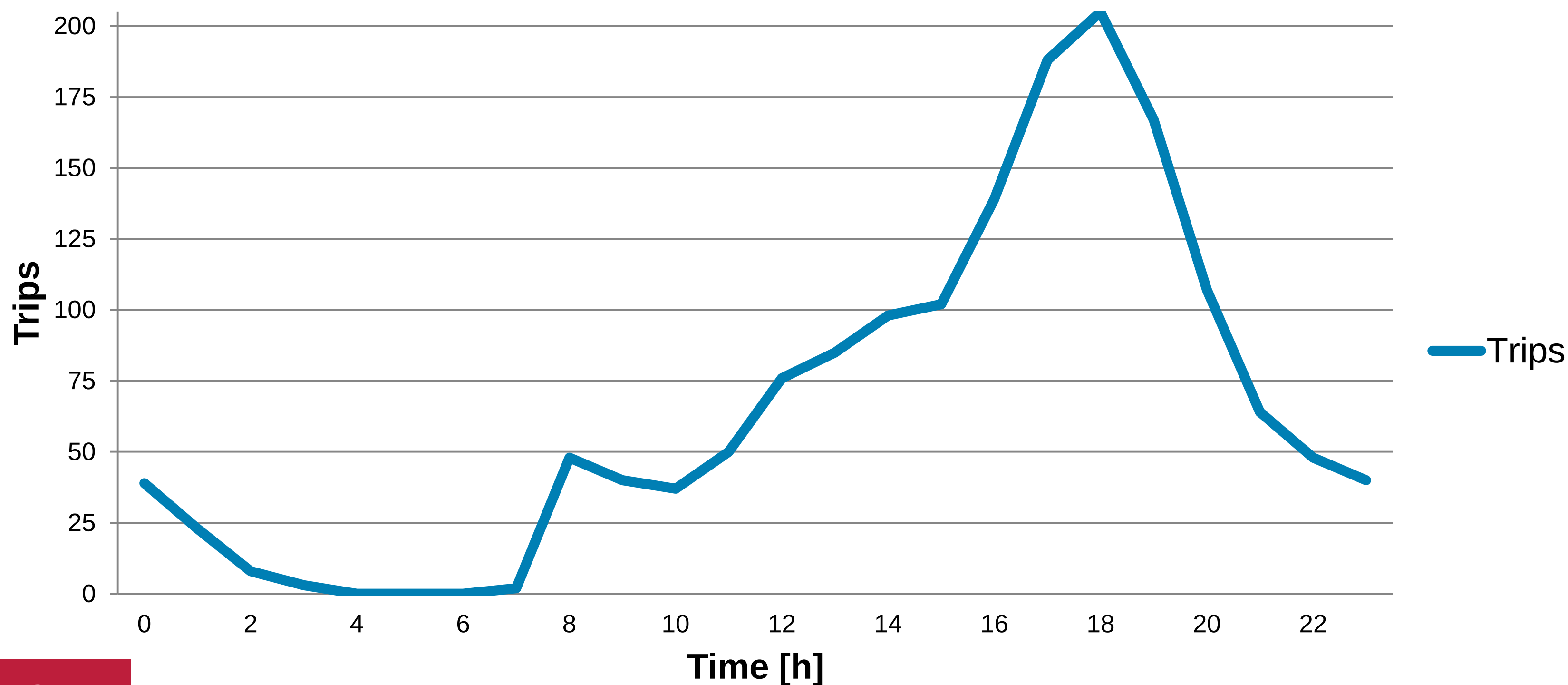
- **Hill Climbing**
  - Chooses the best subsets in the current neighborhood for next iteration
- ⇒ Terminates in a local optimum

- **Simulated Annealing**
  - For further exploitation, chooses randomly subsets from the current neighborhood
    - Accepts (inferior) subsets with probability $\phi \coloneqq \min\left\{1, \exp\left(\frac{o_c - o_n}{T}\right)\right\}$
  - Returns best subsets found
- ⇒ Overcomes local optimality

Case Studies: Instances

- Vienna‘s BSS „City Bike Wien“
  - 59 stations
  - Station capacity of 10-40 bike racks
  - ~1,569 trips per day extracted by Vogel (2016)

Trips in the Course of the Day



Case Studies: Instances

- Vienna‘s BSS „City Bike Wien“
  - 59 stations
  - Station capacity of 10-40 bike racks
  - ~1,569 trips per day extracted by Vogel (2016)

- 24 time periods à 60min

- Target fill levels by Vogel et al. (2014)

- $2, 3, 4, \text{and } 8$ Vehicles

- Vehicle speed of $15\,\frac{km}{h}$

- Vehicle capacity of 10

Case Studies: Results

Algorithm selection:

	Vehicles			
	2	**3**	**4**	**8**
Hill Climbing	211.45	86.09	65.24	57.74
Simulated Annealing	171.99	69.98	**52.77**	49.83

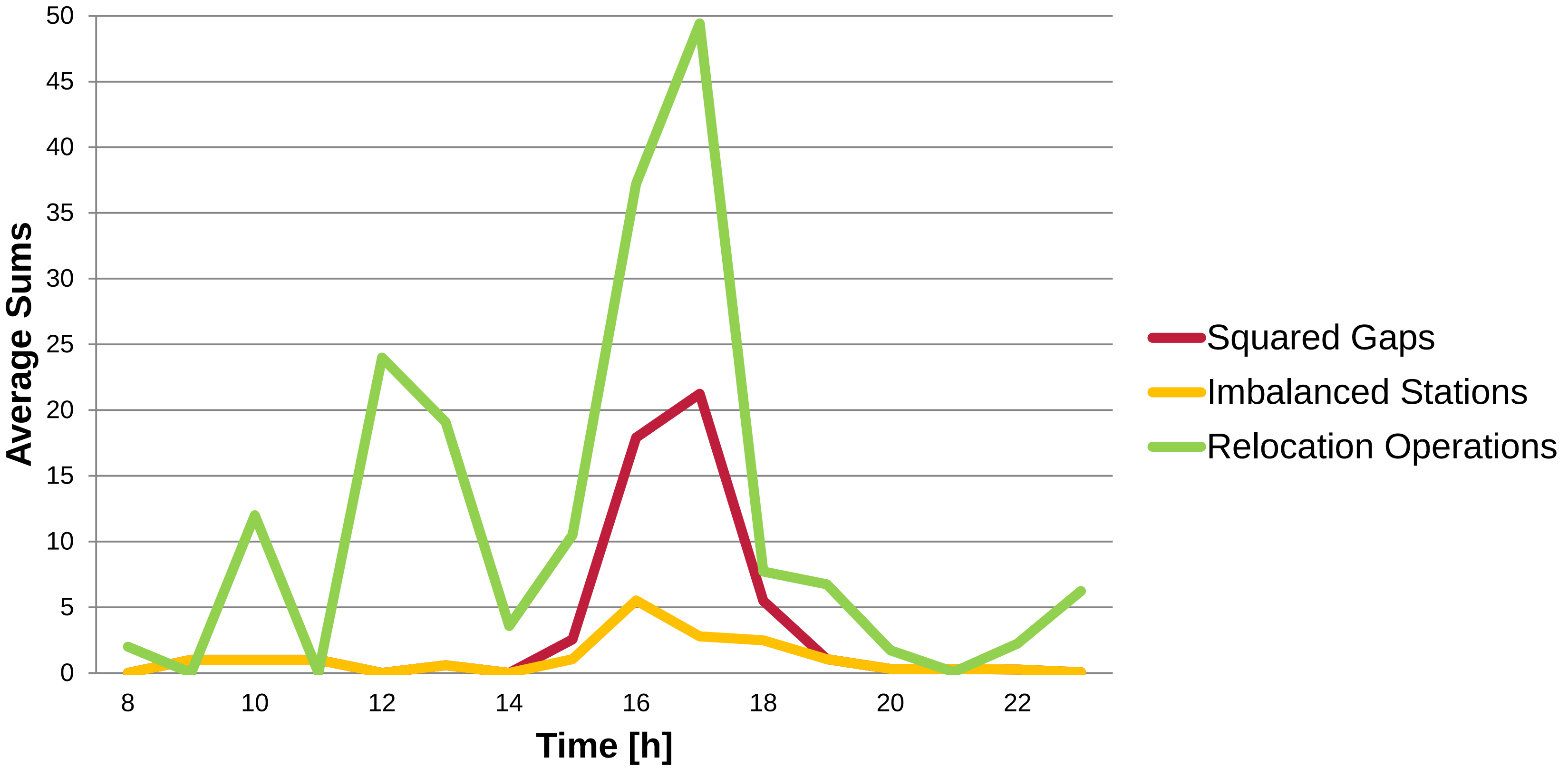
⇒ Simulated Annealing outperforms Hill Climbing.

⇒ Simulated Annealing considering 8 vehicles leads to minor improvements.

⇒ Further analysis of results by Simulated Annealing with 4 vehicles.

Case Studies: Results

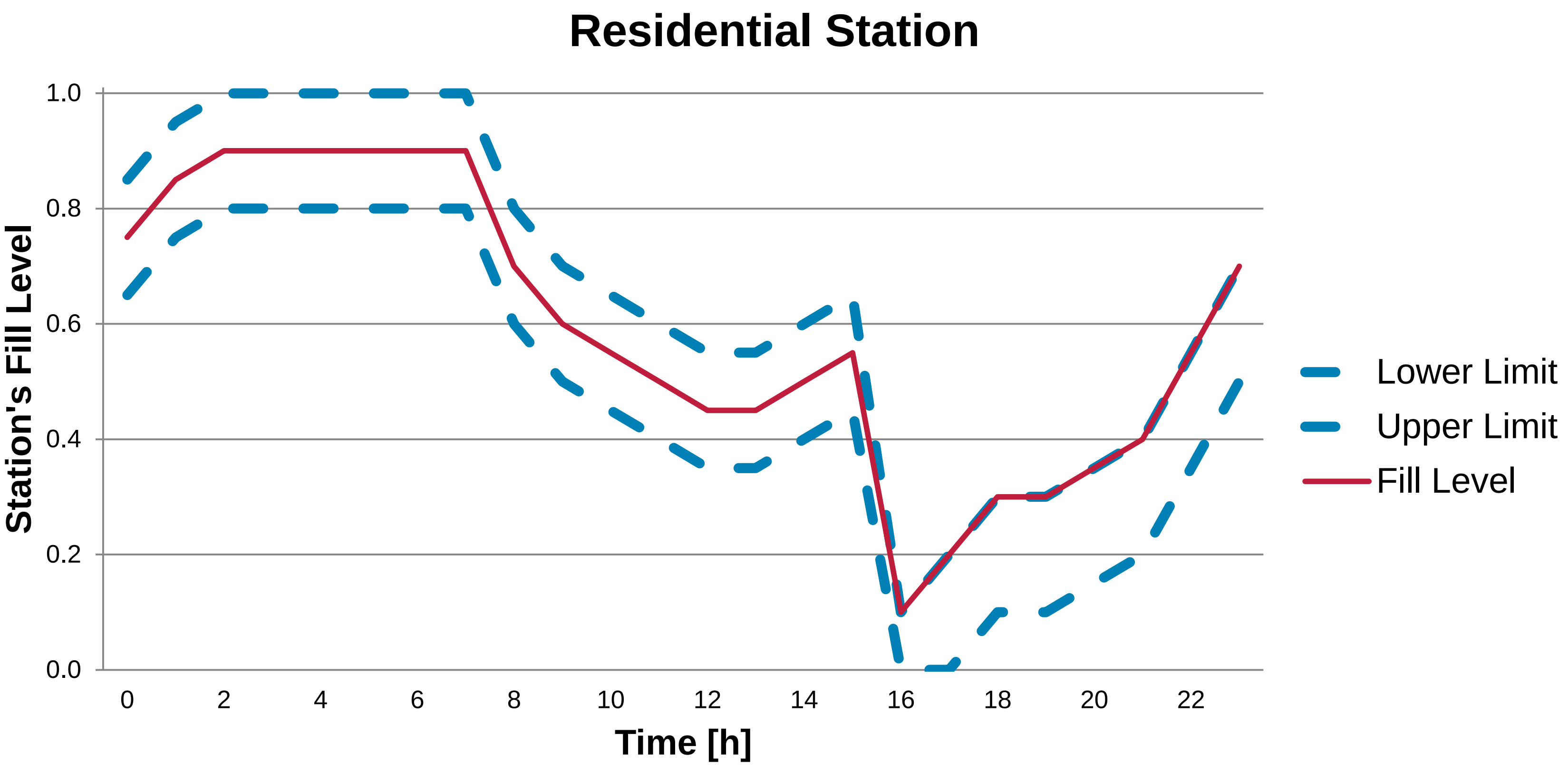
Results for Simulated Annealing and four vehicles:



⇒ Expect for afternoon rushhour stations can be keept balanced.

Case Studies: Results

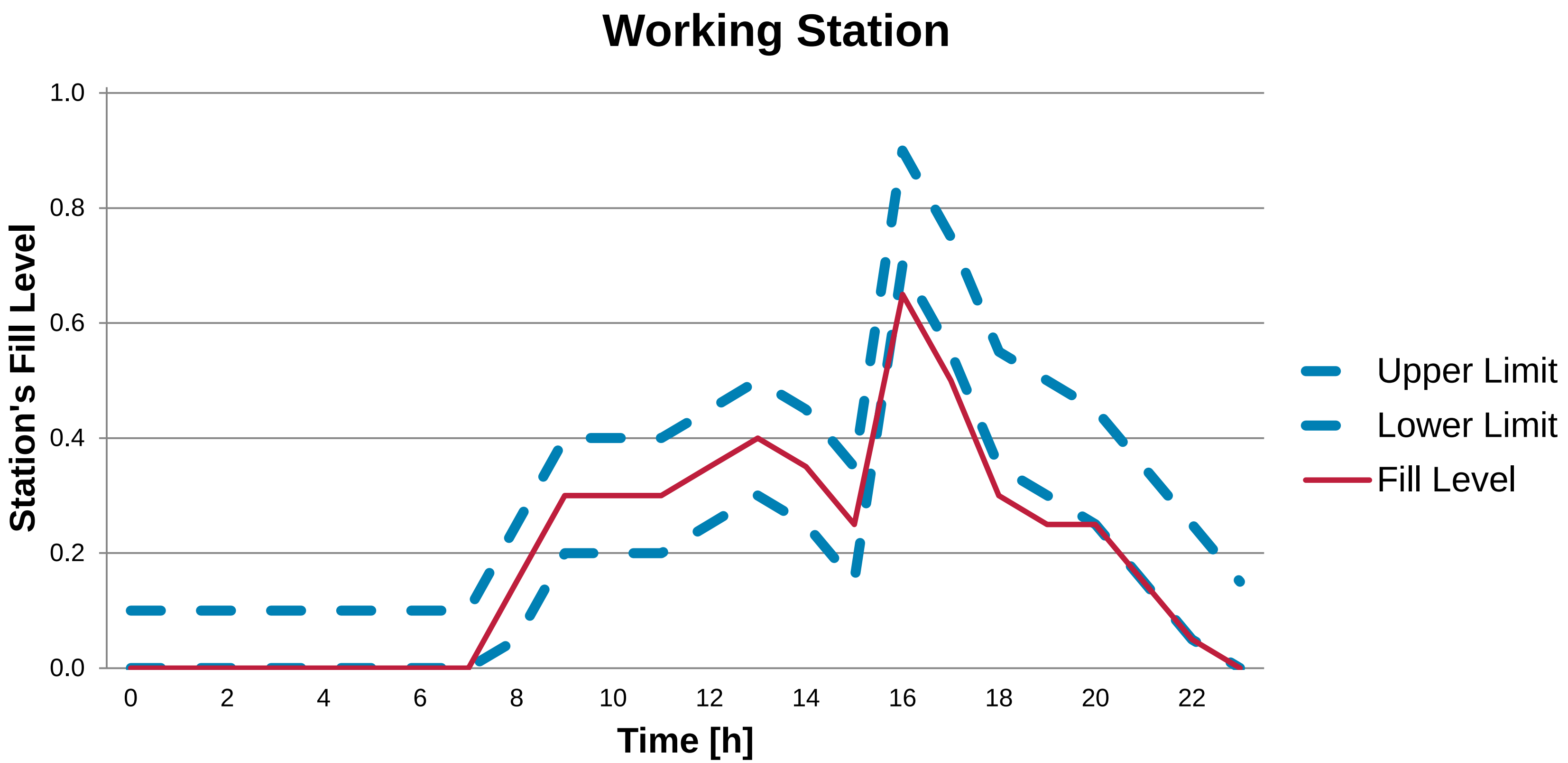
Results for Simulated Annealing and four vehicles:



⇒ Pick-ups before the rushhour.

Case Studies: Results

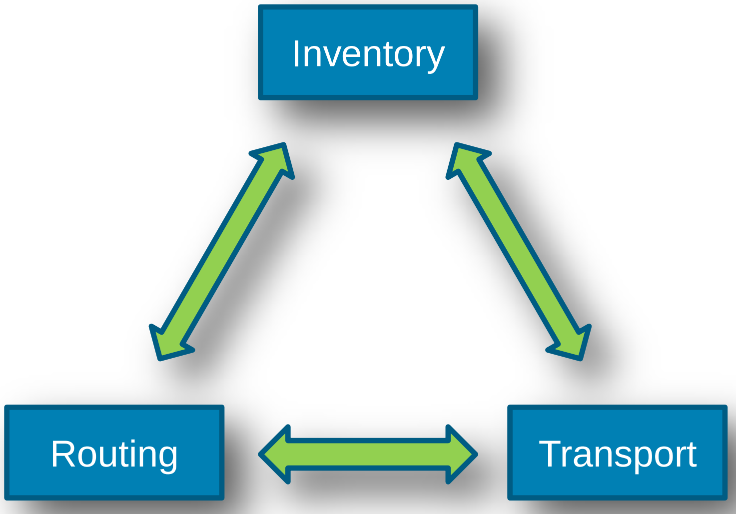
Results for Simulated Annealing and four vehicles:



⇒ Deliveries before the afternoon rushhour.

Summary and Outlook

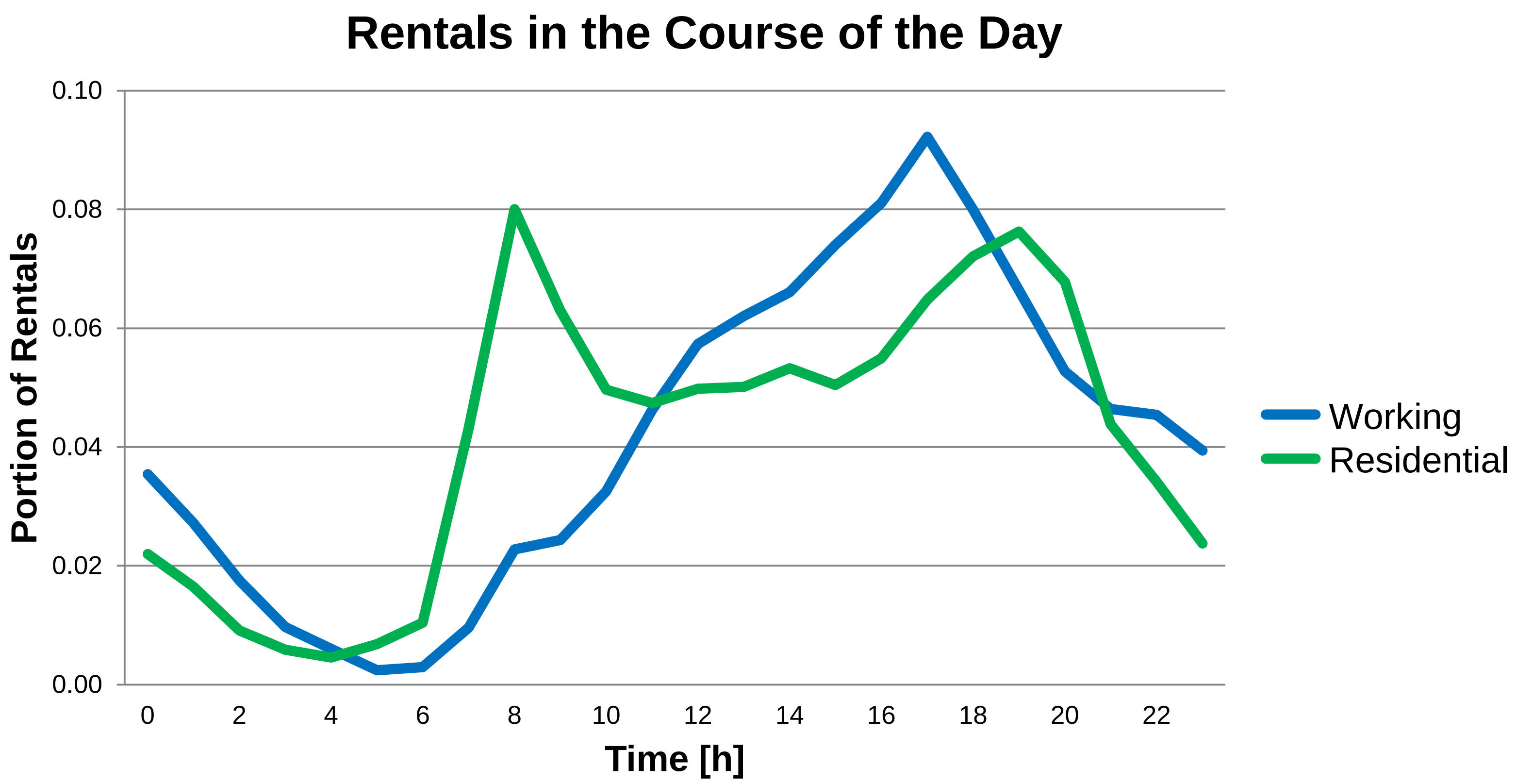
- Inventory Routing Problem
- Goal: realize target fill levels
- Two-dimensional decomposition approach:
  - Solved periods independently
  - Finds subsets allowing efficient rebalancing



- Future research
  - To count failed request directly, evaluate approach in stochastic-dynamic environment.

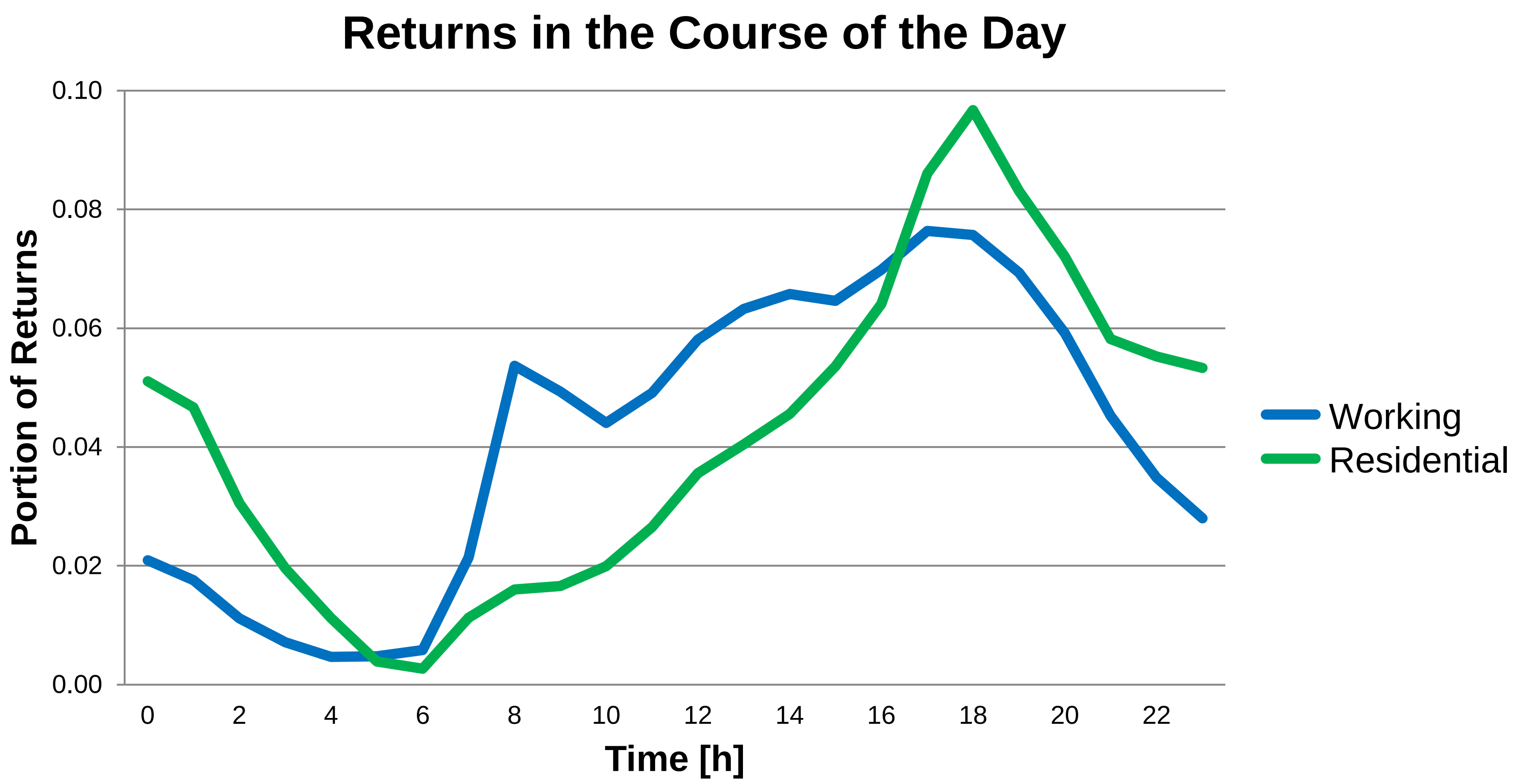
Thank you!

Motivation: Spatio-temporal Variation of Requests



Vogel et al. (2011)

Motivation: Spatio-temporal Variation of Requests



Vogel et al. (2011)

Spatial Decomposition: Decison Making

Decision

Choosing new solutions from the current solutions neighborhood.

- **Hill Climbing**
  - While current solution is no local optimum:
    - Choose the best solution in the current solution‘ neighborhood.
  - Return current solution.
- ⇒ Terminates in a local optimum

- **Simulated Annealing**
  - Initialize T_0.
  - While $\mathrm{T} < \mathrm{T}_{min}$:
    - Choose a random solution in the current solution‘s neighborhood.
    - Accept solution with probability $\phi \coloneqq \min\left\{1, \exp\left(\frac{\mathrm{O}_c - \mathrm{O}_n}{\mathrm{T}}\right)\right\}$.
    - Set $\mathrm{T}_{i+1} \coloneqq c \cdot \mathrm{T}_i$.
  - Return best solution found.
- ⇒ Overcomes local optimality

Case Studies: Results

Operator selection:

		Vehicles			
		2	**3**	**4**	**8**
	no optimization via local search	842.07	754.40	779.96	1,088.18
Hill Climbing	**Insert**	242.10	97.86	71.66	60.34
	Exchange	248.79	113.87	96.61	106.22
	Insert / Exchange	211.45	86.09	65.24	57.74

⇒ No optimization via local search leads to worse results.

⇒ Combination of Insert and Exchange leads to best results.

References

- Vogel P, Greiser T, Mattfeld DC (2011) Understanding bike-sharing systems using data mining: exploring activity patterns. Procedia-Social and Behavioral Sciences, 20:514-523.
- Vogel P, Neumann Saavedra BA, Mattfeld DC (2014) A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems. Hybrid Metaheuristics. Lecture Notes in Computer Science, 8457:16-29, Springer.
- Vogel P (2016) Service Network Design of Bike Sharing Systems – Analysis and Optimization. Lecture Notes in Mobility, Springer.