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

- **Inventory**
  - i.e., fill level
- **Routing**
  - i.e., sequence
- **Transportation**
  - i.e., relocations
Problem Definition: Sets and Functions

- Bike Sharing System
  - Set of stations: $N = \{n_0, ..., n_{max}\}$
  - Capacity: $r: N \rightarrow \mathbb{N}_0$
  - Initial fill levels: $f: N \rightarrow \mathbb{N}_0$
  - Distances: $d: N \times N \rightarrow \mathbb{R}^+$
  - Bikes: $B = \{b_0, ..., b_{max}\}$
  - Planning horizon: $T = \{t_0, ..., t_{max}\}$

- Expected user activities
  - Rental: $R^- = \{r_0^-, ..., r_{max}^-\}$ $r^- = (t, n)$
  - Return: $R^+ = \{r_0^+, ..., r_{max}^+\}$ $r^+ = (t, n)$

- Target Intervals
  - Upper Limits $\bar{\tau}: N \times T \rightarrow \mathbb{N}_0$
  - Lower Limits $\underline{\tau}: N \times T \rightarrow \mathbb{N}_0$

- Optimization
  - Set of vehicles: $V = \{v_0, ..., v_{max}\}$
  - Capacity: $c: V \rightarrow \mathbb{N}$
  - Relocation operations
    - Pickups: $P = \{p_0, ..., p_{max}\}$ $p = (h, n, b)$
    - Deliveries: $D = \{d_0, ..., d_{max}\}$ $d = (h, n, b)$
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 procedure:

1. **Decision**
2. **Operator**
3. **Subsets**
4. **Routing**
5. **Neighborhood**
Spatial Decomposition: Operators

Operators span a neighborhood around a current solution.

- **Insert**
  - Removes one station from its 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 evaluates subsets.

Adapted Nearest Neighbor:

\[ \rho = \frac{\text{gap}}{\text{distance}} \]

\( \text{gap} = 2 \)

\( \rho = \frac{2}{2} = 1 \)

\( \text{gap} = 1 \)

\( \rho = \frac{1}{2} = 0.5 \)

Routing target interval

Adapted Nearest Neighbor:
Spatial Decomposition: Routing

Routing evaluates subsets.

Adapted Nearest Neighbor:

\[ \rho = \frac{gap}{distance} \]

- \( gap = 0 \) for \( n_1 \)
- \( gap = 1 \) for \( n_2 \)
Spatial Decomposition: Decision Making

Choosing new solutions from the current solutions neighborhood.

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

- **Simulated Annealing**
  - For further exploitation, chooses randomly subsets from the current neighborhood
  - Accepts (inferior) subsets with probability $\phi := \min\left\{1, \exp\left(\frac{O_c - O_n}{T}\right)\right\}$
  - Returns best subsets found
  - $\Rightarrow$ 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

Time [h]  | Trips
---------|------
0        | 0
2        | 25
4        | 50
6        | 75
8        | 100
10       | 125
12       | 150
14       | 175
16       | 200
18       | ~225
20       | 200
22       | 150
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, and 8 Vehicles

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

- Vehicle capacity of 10
Case Studies: Results

Algorithm selection:

<table>
<thead>
<tr>
<th>Algorithm Selection</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mountains Climbing</td>
<td>211.45</td>
<td>86.09</td>
<td>65.24</td>
<td>57.74</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>171.99</td>
<td>69.98</td>
<td>52.77</td>
<td>49.83</td>
</tr>
</tbody>
</table>

⇒ 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 rush hour stations can be kept 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

Rentals in the Course of the Day

Portion of Rentals

Time [h]

Vogel et al. (2011)
Motivation: Spatio-temporal Variation of Requests

![Graph showing returns in the course of the day]

Returns in the Course of the Day

- **Portion of Returns**
- **Time [h]**

**Cluster 0**
- **Cluster 2**
- **Working**
- **Residential**

Vogel et al. (2011)
Spatial Decomposition: Decision Making

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’s neighborhood.
    - Return current solution.
  - \( \Rightarrow \) Terminates in a local optimum

- **Simulated Annealing**
  - Initialize \( T_0 \).
  - While \( T < T_{\text{min}} \):
    - Choose a random solution in the current solution’s neighborhood.
    - Accept solution with probability \( \phi := \min\left\{ 1, \exp\left( \frac{O_c - O_n}{T} \right) \right\} \).
    - Set \( T_{i+1} := c \cdot T_i \).
  - Return best solution found.
  - \( \Rightarrow \) Overcomes local optimality
Case Studies: Results

Operator selection:

<table>
<thead>
<tr>
<th>Operator Selection</th>
<th>Vehicles</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>no optimization via local search</td>
<td>842.07</td>
</tr>
<tr>
<td>Hill Climbing</td>
<td></td>
</tr>
<tr>
<td>Insert</td>
<td>242.10</td>
</tr>
<tr>
<td>Exchange</td>
<td>248.79</td>
</tr>
<tr>
<td>Insert / Exchange</td>
<td>211.45</td>
</tr>
</tbody>
</table>

⇒ No optimization via local search leads to worse results.
⇒ Combination of Insert and Exchange leads to best results.
References

