a GRASP and ALNS approach
for bicycle repositioning

Nicholas Vergeylen    Kenneth Sörensen
University of Antwerp
ANT/OR Operations Research Group
antor.uantwerpen.be
nicholas.vergeylen@uantwerpen.be

Pieter Vansteenwegen
KU Leuven Mobility Research Center - CIB, Belgium
University of Leuven, Belgium

TRISTAN 2016
bicycle sharing systems
bicycle sharing systems (BSS)

▶ urban environment
▶ alternative to public transportation
▶ to move from station A to station B
▶ increasingly popular
the problem
repositioning
bike repositioning: 2 decision problems

- approach in literature: BRP
- our approach: we split the problem of making inventory decisions and routing decisions
  - routing problem should be easier to solve
  - human intervention is possible, whereas the BRP is more of a black box

**subproblem 1 [inventory decisions]**
Determining number of bikes to pick up/drop off at which station at what time

**subproblem 2 [routing decisions]**
Routing the repositioning vehicles
our approach

- generate pick-and-drop instructions
- routing
  
intermediate structure
“requests” are pick-and-drop instructions:

<table>
<thead>
<tr>
<th>properties</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>issue time</td>
<td>IT</td>
</tr>
<tr>
<td>id</td>
<td>ID</td>
</tr>
<tr>
<td>station id</td>
<td>STA</td>
</tr>
<tr>
<td>quantity</td>
<td>QUA</td>
</tr>
<tr>
<td>early arrival time</td>
<td>EAT</td>
</tr>
<tr>
<td>late arrival time</td>
<td>LAT</td>
</tr>
<tr>
<td>droptime</td>
<td>DRO</td>
</tr>
<tr>
<td>priority</td>
<td>PRIO</td>
</tr>
</tbody>
</table>
our approach

---

**request generation**

**request scheduling**

---

### request list

<table>
<thead>
<tr>
<th>ID</th>
<th>STA</th>
<th>QUA</th>
<th>EAR</th>
<th>LAT</th>
<th>DRO</th>
<th>PRIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>09:00</td>
<td>09:15</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>10</td>
<td>09:10</td>
<td>09:30</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>-10</td>
<td>08:50</td>
<td>09:00</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>-13</td>
<td>16:30</td>
<td>16:35</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>7</td>
<td>17:00</td>
<td>17:15</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

---
bike request scheduling problem (BRSP)

**data**
- set of requests
- travel times

**decisions**
- determine sequence of requests for each vehicle

**objective**
- minimize the priority-weighted sum of unassigned requests

**constraints**
- vehicle capacity
- (hard) time windows
\begin{align*}
& \min \sum_{i} y_i p_i \\
& \text{s.t.} \\
& \sum_{k} \sum_{n} x_{in}^k + y_i = 1 \\
& z_{ij} \geq x_{jn+1}^k + x_{in}^k - 1 \\
& \sum_{i} x_{i(n+1)}^k \leq \sum_{i} x_{in}^k \\
& \sum_{i} x_{in}^k \leq 1 \\
& x_{in}^k \in (0, 1) \\
& z_{ij} \in (0, 1) \\
& l_k + \sum_{i} \sum_{n' \leq n} x_{in'}^k b_{i} \leq C \\
& l_k + \sum_{i} \sum_{n' \leq n} x_{in'}^k b_{i} \geq 0 \\
& a_j \geq a_i + o_i + t_{ij} - (1 - z_{ij})M \\
& a_i^1 \geq a_i \geq a_i^e \\
& y_i \in (0, 1), l_k \in [0, C], a_i^* \in \mathbb{R}^+ 
\end{align*}
### Request List

<table>
<thead>
<tr>
<th>ID</th>
<th>STA</th>
<th>QUA</th>
<th>EAR</th>
<th>LAT</th>
<th>DRO</th>
<th>PRIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>10</td>
<td>09:10</td>
<td>09:30</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>-10</td>
<td>08:50</td>
<td>09:00</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>-13</td>
<td>16:30</td>
<td>16:35</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>7</td>
<td>17:00</td>
<td>17:15</td>
<td>14</td>
<td>3</td>
</tr>
</tbody>
</table>

### Solution

<table>
<thead>
<tr>
<th>Unscheduled</th>
<th>0</th>
<th>9</th>
<th>18</th>
<th>-</th>
<th>-</th>
<th>-</th>
<th>-</th>
<th>-</th>
<th>-</th>
<th>-</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle 1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Vehicle 2</td>
<td>8</td>
<td>7</td>
<td>15</td>
<td>22</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
1. general solver (GUROBI)
2. GRASP
3. ALNS
ALNS

select D/R combo

adapt selection

execute sequence

evaluate result
select D/R combo

adapt selection

execute sequence

evaluate result
GRASP construction

<table>
<thead>
<tr>
<th>α</th>
<th>request id’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0 10 5 3 8 2 ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>unscheduled</td>
</tr>
<tr>
<td>vehicle 1</td>
</tr>
<tr>
<td>vehicle 2</td>
</tr>
</tbody>
</table>

capacity violation
**GRASP construction**

<table>
<thead>
<tr>
<th>α</th>
<th>request id’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0 10 5 8 2 7 ...</td>
</tr>
</tbody>
</table>

**solution**

<table>
<thead>
<tr>
<th>unscheduled</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>vehicle 1</td>
<td>5</td>
</tr>
<tr>
<td>vehicle 2</td>
<td></td>
</tr>
</tbody>
</table>

OK!
## GRASP construction

### sorted request list

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>request id’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0 10 8 2 7 6 ...</td>
</tr>
</tbody>
</table>

### solution

| unscheduled | 3 - - - - - - - - - - |
| vehicle 1   | 8 5 - - - - - - - - - - |
| vehicle 2   | - - - - - - - - - - - - |

OK!
GRASP construction

- sorts all requests according to a function of request attributes
- groups the top $\alpha$ requests = Restricted Candidate List (RCL)
- tries to schedule a random request in the RCL
  - $\alpha$ is a degree of randomness or diversification of the search
- the constructed schedule is subjected to a local search
request insertion

- sorts all unscheduled requests according to priority
- tries to schedule the best unscheduled request each iteration
- stops when no request could be scheduled during the iteration
### sorted request list

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>request id's</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8 5 2 3 0 ...</td>
</tr>
</tbody>
</table>

### solution

<table>
<thead>
<tr>
<th>unscheduled</th>
<th>10 0 7 6 4 8 ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>vehicle 1</td>
<td>$\triangleright$8 3 0 5 - - - - - - - -</td>
</tr>
<tr>
<td>vehicle 2</td>
<td>2 - - - - - - - - - -</td>
</tr>
</tbody>
</table>

schedule infeasible!
source of infeasibility
GRASP Destroy

source of infeasibility
repair the infeasibility:

- calculate sum infeasibility (SI)
- allow request insertion during construction when it does not worsen SI
results: montreal instance
results: all instances

- antwerp
- barcelona
- bayarea
- daejeon
- dublin
- hamburg
- melbourne
- miami
- montreal
- newyork
- oslo
- paris
- rio
- seville
- shanghai
- taipei
- vienna
- warsaw
- washingtondc

- alns-balanced+random
- grasp-ET
- grasp-LT
- grasp-PRIOR
- grasp-PROFQT
we introduced a new innovative approach for repositioning
practical cases for this model can benefit from heuristics
GRASP-based heuristics incorporates search space characteristics
ALNS improves single GRASP heuristics and creates synergy
a GRASP and ALNS approach for bicycle repositioning

Nicholas Vergeylen  Kenneth Sörensen
University of Antwerp
ANT/OR Operations Research Group
antor.uantwerpen.be
nicholas.vergeylen@uantwerpen.be

Pieter Vansteenwegen
KU Leuven Mobility Research Center - CIB, Belgium
University of Leuven, Belgium

TRISTAN 2016