Intelligent Urban Transportation – Making Last Mile Deliveries more Efficient, Reliable and Sustainable

UMSL, 2015-11-06

Jan Fabian Ehmke
Advanced Business Analytics Group
School of Business & Economics
Freie Universität Berlin

Based on joint work with
Ann Campbell – André Steinert – Barrett Thomas – Tim Urban
Intelligent Urban Transportation

Aim: making urban transportation more efficient, reliable and sustainable

- Individual mobility
  - Sharing of mobility resources
  - Automation leads to increased dynamics of systems

- Transportation of goods
  - Narrow service time windows (→ efficiency & reliability)
  - Minimize emissions (→ efficiency & sustainability)

- Aggregation of operational data
  - Ubiquity of sensors
  - Transactions

- Intelligent data analysis and data mining
- Dynamic and stochastic optimization
Challenges for Urban Deliveries

- Online retailing is the fastest growing retail sector
- **Efficient and reliable** deliveries are critical for lasting success

- This is particularly challenging for **attended deliveries**
  - Examples: grocery, large appliances, repairmen

- Customers expect **on-time deliveries** and narrow service time windows

- Logistics service providers need to find the **optimal trade-off** between efficiency, reliability and sustainability

- Metropolitan areas suffer from **congestion** and **pollution**
Agenda

- Collect, Analyze and Transform Sensor Data
- Set up a Planning Framework for Urban Routing

- Decide Which Customers to Accept
- Find a Balance between Efficiency and Reliability

- Ensure Service Levels in Routing
- Investigate the Theoretical Potential of Reliability

- Consider Sustainability in Routing
- Assess the Importance of Stochastic Travel Times for Minimizing Emissions
Ubiquitous Sensor Data

~230 million Floating Car Data from taxis

Network of
~ 100000 links
~ 128000 nodes

Aerial view with kind permission of GeoContent, Magdeburg
A Planning Framework for Urban Transportation

- **empirical traffic data**
- **Data Mining**
- **information models**
- **digital roadmap**
- **time-dep digital roadmap**
- **modeling of time-dep**
- **abstraction**
- **delivery tours**
- **search**
- **time-dep optimization model**
- **time-dep distance matrix**

**Setup**
- Efficiency
- Reliability
- Sustainability
Time-Dependent Routing

- Optimization of a single route
- The optimal order of customers depends on the departure time

Examples for departure at
- 13:30 → Return at 18:16
- 19:30 → Return at 00:08

- Time-dependent routing follows anticipated congestion of the urban traffic network
Agenda

- Collect, Analyze and Transform Sensor Data
- Set up a Planning Framework for Urban Routing

- Decide Which Customers to Accept
- Find a Balance between Efficiency and Reliability

- Ensure Service Levels in Routing
- Investigate the Theoretical Potential of Reliability

- Consider Sustainability in Routing
- Assess the Importance of Stochastic Travel Times for Minimizing Emissions
Maximize the Number of Customers

- We want to maximize the number of customers we can serve considering **tight time windows**
- How do we decide which customers to accept?
- How do we **balance** profit maximization with the need to provide reliable service?
- Examine several **acceptance mechanisms**
  - Differ in level of information, ease of implementation
- Evaluate them using **simulation framework**
  - Simulated demands
  - Simulated travel times to reflect congestion in metropolitan areas
Customer Acceptance – Related Work

- **Selection** of a service time window
  - Maximize profits by accepting as many customer requests as possible while ensuring service quality
  - Decide on when to open and close time windows
  - Assumed deterministic travel time
  - Not look at impact of congestion/time dependency

- **Vehicle routing** and scheduling
  - Minimizing costs of delivery
  - Consider time-dependent travel times
    - time-dependent VRPTW
  - Ichoua et al. 2003, Eglese et al. 2006, Fleischmann et al. 2004, Maden et al. 2010
Home Delivery Problem – Problem Setting

- Retailer offers a **predefined set of time** slots on a day
- Delivery requests arrive before start of service

- Customer selects time slot
  - Each customer has first and second choice
  - Provide first choice if available
  - Provide second choice if first choice not available
  - Customer leaves if neither is available

- Need to quickly decide if a request can be handled in a particular time slot

- Objective: **maximize the number of accepted requests**
  (assumption: vehicle capacity not binding)
Home Delivery Problem – Solution Approach

- Need to quickly decide if a request can be handled in a particular time slot

- This can be done in a rough/approximate way
  - Rules of thumb
  - Fast/easy
- Refer to as **static approaches**

- This can be done in a more dynamic way
  - Use more detailed information, build routes
  - Can accept more deliveries
  - May accept too many deliveries!
  - Congestion, stochastic travel times
- Refer to as **dynamic approaches**

<table>
<thead>
<tr>
<th>Time Slot</th>
</tr>
</thead>
<tbody>
<tr>
<td>[12:00,12:59]</td>
</tr>
<tr>
<td>[13:00,13:59]</td>
</tr>
<tr>
<td>[14:00,14:59]</td>
</tr>
<tr>
<td>[15:00,15:59]</td>
</tr>
<tr>
<td>[16:00,16:59]</td>
</tr>
<tr>
<td>[17:00,17:59]</td>
</tr>
<tr>
<td>[18:00,18:59]</td>
</tr>
</tbody>
</table>
Dynamic Approaches (1/2)

- **Dynamic (DYN)**
  - Incorporate time-dependent travel times and precise routing
  - Consider a time-dependent routing solution for feasibility check (I1/Solomon 1987)
  - If the current request (and the already accepted requests) can be accommodated feasibly, the request is accepted
  - This can still fail due to stochastic travel times!

- **Dynamic Simple Buffer (DYN-SBF)**
  - Include a simple buffer time $bt$ in the feasibility check of each request
  - Accept a request $i$ only if $arr_i + bt < late_i$
  - E.g. get there 10 minutes before window closes
  - All customers have the same buffer
Dynamic Approaches (2/2)

- **Dynamic Buffer (DYN-BUF)**
  - Accept a request $i$ only if $arr_i + bt_i < late_i$
  - Computing this buffer correctly is non-trivial!

- Idea: recursive calculation of variance that depends on
  - Proportion $\beta_i$ of arrival time distribution after $early_i$
  - Variation of travel times $\sigma_{i-1,i}^2$
  - $y_i = \text{cumulated standard deviation of arrival time}$
    - $y_i = \sqrt{\beta_{i-1}y_{i-1}^2 + \sigma_{i-1,i}^2}, y_0, \beta_0 = 0$
    - $bt_i = \alpha \beta_i y_i$
      (with user specific service level $\alpha$)
Experimental Design

- Real-world inspired urban road network
- Time-dependent travel times generated from speed multipliers
  - To reflect daily congestion patterns
- Two zones of customer locations
  - Inner city and suburbs
- Random generation of customer requests with random selection of time windows
  - 100 instances from each zone
  - Simulated 1000 times
- Basic test
  - Time windows width of 60 minutes
  - Service time of 20 minutes
  - Three vehicles
Impact of DYN-BUF

TW 15:00-16:00
DYN: Arrival 15:56
BUF: Arrival 15:32

TW 15:00-16:00
DYN: Arrival 15:59
BUF: Arrival 15:51

TW 17:00-18:00
DYN: Arrival 17:46
BUF: Arrival 17:35

TW 18:00-19:00
DYN: Arrival 18:58
BUF: Arrival 18:14

DYN: 34 customers
Proportion served late: 94%
BUF: 34 customers
Proportion served late: 7%
Suburban Delivery

- DYN-SBF accepts the least requests with 10 minute buffer
- DYN accepts the most but is late in 89% of realizations
- DYN-BUF: slightly smaller request acceptance, but occurrence of lateness is reduced a lot
- SLOT accepts far fewer but is also rarely late – tends to be conservative
Agenda

- Collect, Analyze and Transform Sensor Data
- Set up a Planning Framework for Urban Routing

- Decide Which Customers to Accept
- Find a Balance between Efficiency and Reliability

- Ensure Service Levels in Routing
- Investigate the Theoretical Potential of Reliability

- Consider Sustainability in Routing
- Assess the Importance of Stochastic Travel Times for Minimizing Emissions
We aim to minimize costs of delivery and ensure a given service level for all customers (“chance constraint”)

Related work mainly focuses on recourse, i.e., arrivals before and after service time windows are penalized (“soft time windows”; Taniguchi et al. (2001), Ando and Taniguchi (2006), Tas et al. (2013a), Tas et al. (2013b))

Carefully consider how to compute arrival time distributions for each customer

Enhance feasibility check such that it can be “plugged” into any algorithm for the VRPTW
Service Levels – Problem Setting

- Based on standard VRPTW formulation: One depot with a set of identical vehicles; each customer has a service time window \([early_i, late_i]\)

- Travel time from customer \(i\) to customer \(j\) is stochastic (but not time-dependent)
  - We first assume normally distributed travel times
  - Travel times are statistically independent
  - We are particularly interested in distributions of the arrival time \(AT_i\) and start of service time \(ST_i\)

- Objectives
  - compute tour plan with (1) minimum number of vehicles and (2) minimum total tour duration;
  - ensure that the probability of arriving at each customer \(i\) by \(late_i\) is greater or equal to a given service level \(\alpha\), i.e., \(P(\text{late}_i) \geq \alpha\).
Propagation of Arrival Times

- To verify a certain probability of arriving at each customer $i$ by $late_i$, we must know the distribution of $AT_i$

- The expected arrival time at a customer $i$ is based on previous travel times and service time windows:
  - In case of no waiting on the route to $i$: means and variances can be summed according to the convolution of the normal distribution
  - In case of waiting with high probability: start of service will be at $e_{i-1}$; variance at customer $i$ will be due to the travel time between $i-1$ and $i$
  - In case of relatively low probability of waiting: carefully consider how this affects the distribution of $ST_{i-1}$
    → combine start of service time and travel time distributions
Extending Feasibility Checks for the VRPTW

- We enforce

\[ P(AT_i \leq late_i) \geq \alpha \]

based on the particular arrival time distribution generated by extreme value theory

- More formally, for each customer \( i \), we guarantee

\[ \mu_{AT_i} + z\alpha \sigma_{AT_i} \leq late_i \]

- We embed the feasibility check and the computation of arrival and start-service times in a tabu search algorithm (LANTIME, Maden et al. (2010))

- We adapted the way LANTIME handles infeasible solutions
Computational Experiments (1/2)

- Computational experiments based on Solomon benchmark data sets
  Mean travel time is based on Euclidean distance with CoV = 20% or CoV = 40%
- We consider three different service levels ($\alpha = 50\%, 84.13\%, 97.73\%$)

![Computational Experiments Diagram]

Intelligent Urban Transportation
Computational Experiments (2/2)

- R1: randomly distributed customers, routes with few customers, tight service time windows

<table>
<thead>
<tr>
<th>num</th>
<th>CoV</th>
<th>$z_a$</th>
<th>vehicles</th>
<th>% incr</th>
<th>time</th>
<th>% incr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0.2</td>
<td>0</td>
<td>9.2</td>
<td>–</td>
<td>874.1</td>
<td>–</td>
</tr>
<tr>
<td>50</td>
<td>0.2</td>
<td>1</td>
<td>10.0</td>
<td>9%</td>
<td>899.0</td>
<td>3%</td>
</tr>
<tr>
<td>50</td>
<td>0.2</td>
<td>2</td>
<td>10.6</td>
<td>15%</td>
<td>967.5</td>
<td>11%</td>
</tr>
<tr>
<td>50</td>
<td>0.4</td>
<td>0</td>
<td>9.6</td>
<td>–</td>
<td>873.4</td>
<td>–</td>
</tr>
<tr>
<td>50</td>
<td>0.4</td>
<td>1</td>
<td>11.0</td>
<td>15%</td>
<td>973.7</td>
<td>11%</td>
</tr>
<tr>
<td>50</td>
<td>0.4</td>
<td>2</td>
<td>12.8</td>
<td>33%</td>
<td>997.5</td>
<td>14%</td>
</tr>
<tr>
<td>100</td>
<td>0.2</td>
<td>0</td>
<td>15.6</td>
<td>–</td>
<td>1386.5</td>
<td>–</td>
</tr>
<tr>
<td>100</td>
<td>0.2</td>
<td>1</td>
<td>17.8</td>
<td>14%</td>
<td>1410.8</td>
<td>2%</td>
</tr>
<tr>
<td>100</td>
<td>0.2</td>
<td>2</td>
<td>18.4</td>
<td>18%</td>
<td>1515.9</td>
<td>9%</td>
</tr>
<tr>
<td>100</td>
<td>0.4</td>
<td>0</td>
<td>16.0</td>
<td>–</td>
<td>1495.7</td>
<td>–</td>
</tr>
<tr>
<td>100</td>
<td>0.4</td>
<td>1</td>
<td>18.6</td>
<td>16%</td>
<td>1504.0</td>
<td>1%</td>
</tr>
<tr>
<td>100</td>
<td>0.4</td>
<td>2</td>
<td>20.0</td>
<td>25%</td>
<td>1620.3</td>
<td>8%</td>
</tr>
</tbody>
</table>
Agenda

- Collect, Analyze and Transform Sensor Data
- Set up a Planning Framework for Urban Routing

- Decide Which Customers to Accept
- Find a Balance between Efficiency and Reliability

- Ensure Service Levels in Routing
- Investigate the Theoretical Potential of Reliability

- Consider Sustainability in Routing
- Assess the Importance of Stochastic Travel Times for Minimizing Emissions
Consider Sustainability – Minimize Emissions

- Increasing transportation activities induce higher emission rates (→ greenhouse gas emissions)
- Focus of routing is on distances & travel times
- How different are distance and travel-time based routes compared to emissions-minimized routes?
- Include emissions objectives, esp. load
- Consider detailed information on time-dependent speeds from historical speed database
- Find a computationally tractable way to adapt vehicle routing heuristics
Minimize Emissions – Problem Setting

- Based on a standard VRP setting
  - Given fleet of vehicles
  - Customers with pickups of a particular weight

- Emissions are modeled using the Comprehensive Emissions Model (CEM) (Barth et al. 2005)

- Load will vary on the paths between the customers
  - What is the emissions optimal path for a given load and departure time?
  - Paths cannot be precomputed in general!
Modeling Emissions

- Emission is a function of speed, distance, and weight of a vehicle

- **CEM** modified for time-dependent fuel consumption on arc $a$ at time $t$ and load $l$

  $$F_a(t, l) = E \left[ \lambda \left( k N_e V \frac{d_a}{v_{a,t}} + \gamma d_a v_{a,t}^2 + \gamma \alpha (\mu + l) d_a \right) \right]$$

  - With $d_a$ as distance, $V_{a,t}$ as speed, and $l$ as load
  - $\lambda$, $k$, $\gamma$, $\beta$, and $\alpha$ as engine specific parameters

- Relationship between speed and emissions is **non-linear**

- Optimization based on **mean speeds** underestimates emissions (Jensen’s Inequality $f(\text{mean } V_e) \leq \text{mean } f(V_e)$)
Related Literature

- Is based on different emissions models (popular: CEM and MEET).

- Often assumes that vehicles can drive with emissions-minimizing speed (Bektas and Laporte (2011) and Demir et al. (2012)).

- Considers congestion by means of time-varying travel times and speeds (Figliozzi (2010), Jabali et al. (2012), Franceschetti et al. (2013)).

- Also related: VRPs considering congestion charge (Wen et al. (2014), Qian and Eglese (2014)).
Intelligent Precomputation

- Vehicle routing heuristics usually require the **precomputation** of expected emissions costs
  - between every pair of customers
  - at each departure time
  - for every possible load

- Precomputation is conducted by **shortest path algorithms**
  - How to handle varying load sizes?

- **Proposition**: If a path between a pair of nodes is optimal in terms of emissions for an empty truck as well as a full truck, the path is optimal in terms of emissions for all load sizes in between.
Value of Intelligent Precomputation

![Graph showing the proportion of precomputed paths over time of day.

- Black line represents 6350 kg.
- Gray line represents 12700 kg.

The graph illustrates the efficiency of precomputation across different times of the day.]
Routing Algorithm

- We employ a well-known **tabu search** solution approach (LANTAIME; Maden et al. 2010)
- We adapt standard tabu search neighborhoods
- An initial solution is constructed with $I_1$ (Solomon 1987)

- Primary objective: minimize the **number of vehicles**
- Secondary objective: minimize the **total expected emissions costs**

- **Lookup** precomputed, load-independent emissions costs and add load-dependent costs when load is known
- If lookup fails, compute and store emissions cost for a particular load **online**
Computational Setup

- Four **departure times** on Tuesdays (06:30, 12:30, 15:30, 19:30)
- Five **test sets** with 10/30 customers each (inner city, suburban, mixed)
- Two different **types of vehicles** (standard, heavy)
- **Heterogeneous load**: select three customers whose load totals 90% of capacity (random, farthest, closest)
- **Homogeneous load**: divide the space evenly between the customers
- **Multiple vehicles**: compare results for 30 customers on different fleets (standard, heavy, mixed)
TSP: Examples for Homogeneous Loads

- **TT**: clockwise, total length: 73.71 km, total emissions: 17.63 kg CO$_2$
- **EM-LI**: avoids highways, total length: 66.75 km, total emissions: 15.55 kg CO$_2$
- **EM-LI**: counterclockwise, total length: 66.10 km, total emissions: 15.40 kg CO$_2$
TSP: Examples for Heterogeneous Loads

- EM-LI route serves heavy customers early in the tour
- EM-LD route: heavy customers are shifted to the end
- 4 kg Savings in CO₂
Examples for Heterogeneous Fleet, 15 Closest Heavy (Heavy Vehicle only)

- Highest savings with mixed fleet and heterogeneous customers
- Mixed instances/closest 15 heavy: savings in emissions of 10.25% vs. DIST routes
Summary and Outlook

- It is important to consider **dynamics of urban traffic** in transportation optimization.

- Finding the **right level of aggregation** is crucial for run time and quality of solutions.

- Smart ideas of **incorporating real data into optimization** can help making planning of routing more efficient, reliable and sustainable.

- Consider more realistic **customer behavior**, different **emissions models** and more efficient ways of including **detailed traffic and customer data** into optimization.
Thank you for your attention!

Further reading:


