Carpooling Model

Luk Knapen

Universiteit Hasselt IMOB

June 14, 2013

IMOB
Overview

Transportation Research Institute

1. Traffic Safety (20 people)
2. Mobility - Travel Behavior (15 people)
3. Multi-disciplinary research institute (Faculty of Economics)
4. PhD subjects and projects: Mobility
    1. Mode selection - Effect of Public Transportation Level of Service
    2. Accessibility via public transportation
    3. Mobility problems for MS-Patients
    4. Schedule (daily agenda) prediction
    5. Electric Vehicles and Smartgrids
    6. Cooperative scheduling
    7. Use of big data sources for schedule prediction

Schedule Prediction

- Multimodal tours
- Constraint based scheduling

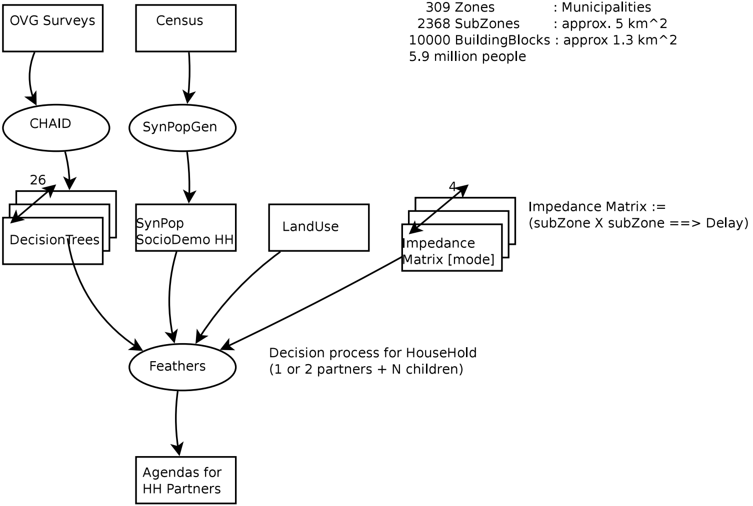
Modeling Cooperation

- Carpooling as an example

Schedule Adaptation - ReScheduling

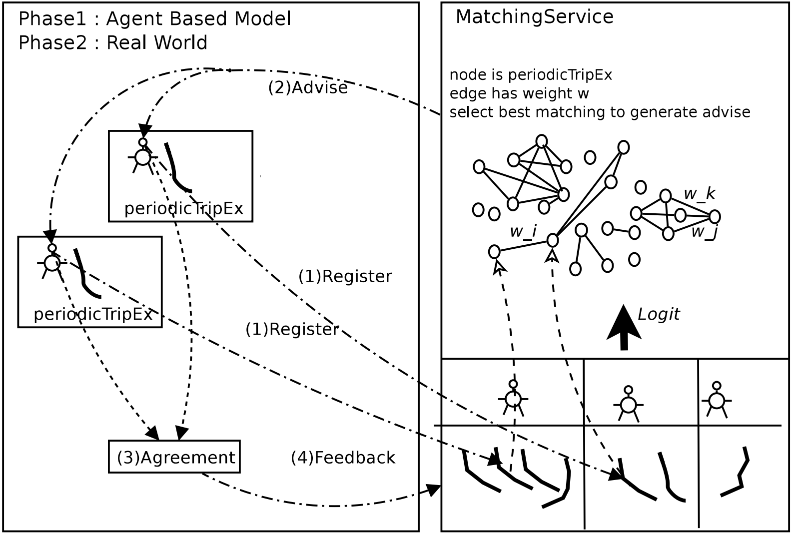
- Due to unexpected events (WIDRS)
- EV charging with time dependent electricity price
- In order to carpool

Context
FEATHERS



309 Zones : Municipalities
2368 SubZones : approx. 5 km^2
10000 BuildingBlocks : approx 1.3 km^2
5.9 million people

Context
Testing Global Matcher



Impedance Matrices

Definition

- TAZ (Traffic Analysis Zones) to model traffic
- ImpedMatrix specifies travel time for given time-of-day for each OD-pair

Calculation

- TransCAD SUE (Stochastic User Equilibrium) is used
- Very time consuming (12[min] ... 16[min])

Assumption : Level of carpooling does *almost* not affect travel times

- Precalculate Impedance Matrices
- Replace 96 IM by spline approximation for each OD-pair

Carpooling Candidate Social Network

Network Building

1. Sample number of acquaintances A from a *power law* distribution
2. Use parameters found in *big data analysis* papers
3. Sample fraction $f_W \sim normal(\mu_W, \sigma_W)$ of people sharing *work* location
4. Sample home-home distance for $A \cdot (1 - f_W)$ people
    1. using power law distribution
    2. using GSM call data in Belgium
5. Sample $A \cdot (1 - f_W)$ people from synthetic population at required distance

Carpooling Candidate Social Network

Exploration

- Initiator contacts direct neighbours only
- Pool can consist of people not directly related before

Evolution

- Every carpool partner becomes directly linked forever
- No other evolution in first implementation (focus on limited set of mechanisms)

Link Strength (LS)

- Used for *reputation propagation* only
- ILS : Initial LS sampled from cut-off exponential distribution
- CPLS : Carpooling induced LS : grows with number of trips co-driven
- AS : Actual LS $= \max(ILS, CPLS)$

Time Scale

Model Evolution

- Time is relevant w.r.t. gossip, feedback, info propagation, ...
- No micro-simulated traffic
- Time resolution : 1 (working) day
- Simulation period : couple of years

Person (Individual) Attributes Fixed Initial

Person

- HH_SEC : socio-economic category (income category)
- P_Age : age category
- P_Gender : gender

Car

- carCategory : car size indication
- carCapacity : number of seats (driver included)
- carMarginalCost : unit distance cost

Person (Individual) Attributes
Fixed Initial

For pre-negotiation filtering

- $s_{max}[k]$: maximal savings when $k \leq carCap$ partners participate

Open questions for *Person*

- `ucLow` : lower limit for *relative utility/cost* (never accept below this limit)
- `ucReconsider` : limit for *relative utility/cost* that triggers pool membership re-consideration/re-avaluation. Relative utility

Person (Individual) Attributes
Fixed Initial

Schedule (for each working day)

- Episode := (trip,activity)
- Schedule := List of Episode (predicted by FEATHERS)

Derived Schedules

- A schedule S is derived from S_{orig} by *reSequencing*
- *shopping activities* only can be moved
- SubSequence of non-shopping activities is identical in S_{orig} and all derivatives S

ScheduleWithTimeWindows

- Schedule s;
- Function giving *departure time window* for return trip for each valid departure time for *home-work* trip
  $F : \mathbb{R} \to \mathbb{R} \times \mathbb{R} : t \in [dt_{HW}^{early}, dt_{HW}^{late}] \mapsto (dt_{WH}^{early}(t), dt_{WH}^{late}(t))$

Person (Individual) Attributes
Fixed Initial

Daily Alternatives

- `DailyAlts := Set of ScheduleWithTimeWindows`
- Contains original FEATHERS Schedule S_{orig}
- Utility $U(S_{orig}) = 1$

Weekly Alternatives

- Array for each working day : `WeeklyAlts := DailyAlts[workDay]`

Person (Individual) Attributes Evolving - Affected by Carpooling

Person Status

- `poolMemberships` : *actuallyCarpooling*, *carpooledBefore*
- `negotiating` : Person can participate in one negotiation at a time

Reputation Indicators

- `safetyReputation` : Reputation as a driver
- `timelinessReputation` :
  - Associated with (TripExecution,Person) tuple
  - Meaning: *(Almost) Always on time (as driver, passenger)*
- *Reputation* not used in first experiments (in order to evaluate/validate other mechanisms first)

Predicting Negotiation Outcome

Motivation

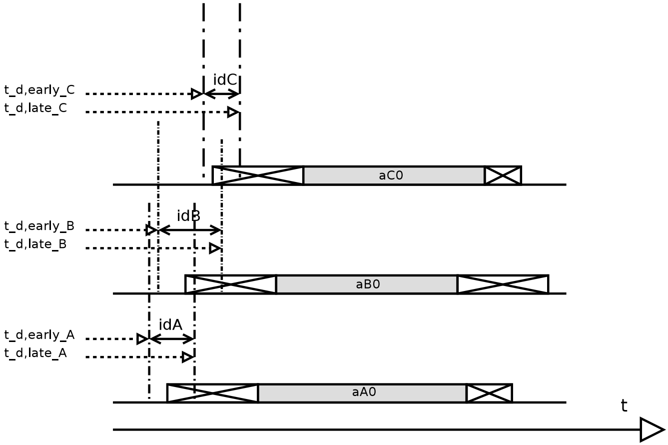
- Negotiation is expensive (computations, object creation)
- Try to filter unrealistic proposals as effective/efficient as possible
- Use *learning* module to predict negotiation outcome
- Use `logit` predictor (logistic regression)
- The predictor represents common knowledge (not individual)
- Initially predictor is *ignorant* and always predicts success: (*false positives* can be detected, *false negatives* cannot)

Predicting Negotiation Outcome

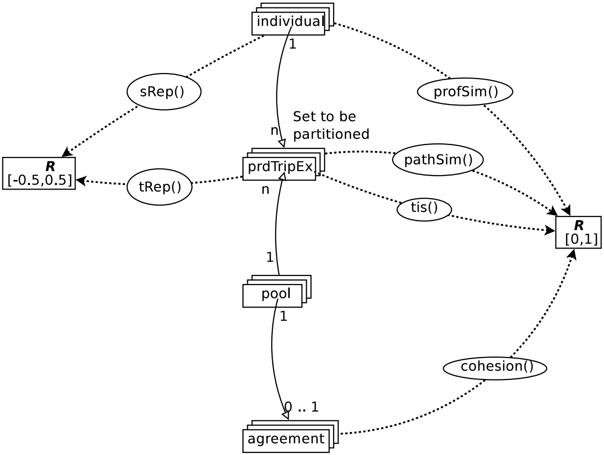
Inputs

1. Person pair characteristics
   1. `profSim` : age, gender
   2. `safetyReputation` : from feedback by passengers
   3. `timelinessReputation` : from feedback by passengers
2. Trip pair characteristices
   1. `pathSim` : path similarity (how does driver's carpooling path differ from solo driving path ?)
   2. `timeSim` : time similarity (how long is period in which to start cooperation) ?

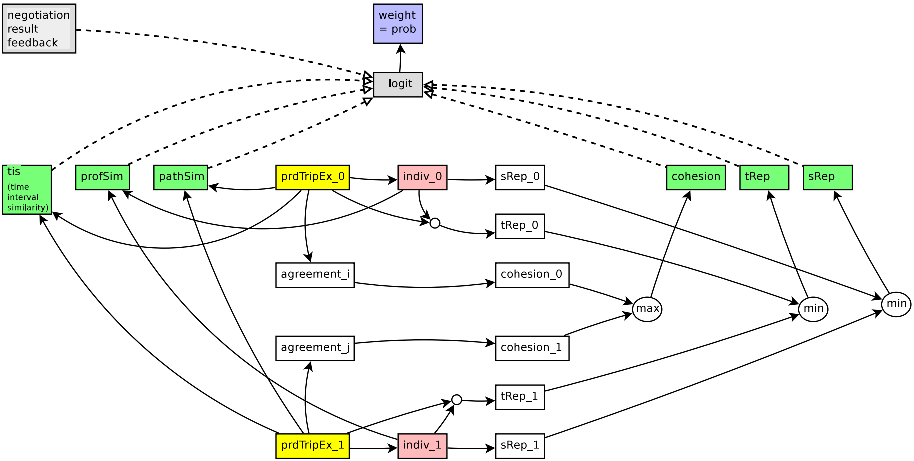
Predicting Negotiation Outcome



Predicting Negotiation Outcome



Predicting Negotiation Outcome



Predicting Negotiation Outcome

Multiple Predictors

1. Specific predictor for each pool size
2. Inputs are sums and products of pairwise similarity values
    1. $S_p = \sum_{i_0, i_1 \in p, i_0 \neq i_1} s^m(i_0, i_1)$
    2. $P_p = \ln(\prod_{i_0, i_1 \in p, i_0 \neq i_1} s^m(i_0, i_1))$

Utility Maximizing Individuals

Assumptions

- Individuals maximize *utility* (difficult concept)
- Marginal utility is monotonically decreasing with duration
- $\frac{\partial U}{\partial d} = k \cdot e^{-\alpha \cdot d}$
- `FEATHERS` predicted schedule is optimal
- Utility reaches 0.95 for longest activity
- $k_a \propto \frac{1}{d}_a$
- Utility $= 1$ for `FEATHERS` schedule (allows to compare *pool* utilities)

Utility Maximizing Individuals

MASWOIC

- Maximal Activity Sequence WithOut Internal Constraints

Time Pressure

- Defined as $\frac{U_0}{d}$ (original utility per time unit for activity)
- Relative time pressure has upper limit (higher fro shorter activities)

Utility as a Function of Time

- Calculate utility for trip start at $t_0 \pm k \cdot \Delta t$ (creates MASWOIC's)
- Time limits (windows) derived from
  - Maximal acceptable $\frac{\Delta U}{U}$ from maximal savings s_{max}
  - Maximal acceptable *relative time pressure*

Trip Cost

Cost components

- Account for distance related cost only (no payment for extra time)
- Account for marginal cost (no investment, insurance, . . .)
- Cost calculated from car specific consumption
- Yearly cost for commuting trip is compared to net household income
- Figures (numeric values) taken from Belgian statistics

$\frac{\Delta U \cdot C}{U \cdot \Delta C}$

Use relative value to avoid *utility/monetary* problem

1. Value of time (VOT) data are difficult to find
2. VOT has been investigated for *in-vehicle-time* but not for activities
3. Assumption : people are capable to evaluate $\frac{\Delta U}{U} \leq \frac{\Delta C}{C_0}$
4. People have a feeling about ratio between
    1. relative utility
    2. marginal travel cost relative to net income
5. Same idea applied for *EV-charging simulator under dynamic pricing*

Becoming an *initiator*

1. Out of the blue (small probability)
2. By perceiving change in environment
   1. Utility drop because someone leaves
   2. Failed negotiation attempt: invite non-selected individual

Augment Schedules by adding Constraints

- Shopping activities within opening times
- Bring/Get (drop/pick) activities within small time window

Augment Schedules by adding Constraints

- Constraints introduce *MASWOIC*s
- Utility optimisation by relaxation algorithm

Determine for each candidate participant

1. Original schedule
2. Variants by *reSequencing*
3. For negotiation, a schedule can be represented by
    1. two trips (*HW* and *WH*)
    2. associated start time intervals
    3. $U(t)$ for start intervals

Submit schedules to pool

Problem

- Find a driver
- Find a route
- Consider for each participant at most one submitted schedule
- Maximise pool size

Algorithm

- Enumerate solutions
- Predict negotiation outcome
- If predicion is positive then negotiate
- Feedback negotiation result

Finding pools

```
foreach size // decreasing
{  foreach subset(size) of participants
   {  foreach driver
      {  for each passenger pick/drop sequence
         {  for each pair of participants
            {  calculate pairwise profSim
               calculate pairwise pathSim
               calculate pairwise timeSim
            }
         }
         if (logitPredictsSuccess() negotiate();
      }
   }
   if poolsFormed() break;
}
findBestPool();
```

Negotiation

Evaluate possible start times

- Consider discrete time values in time windows intersection
- Fetch utility for each participant supplied from $U(t)$
- Keep solution delivering maximal total utility