A Delay-Robust Touristic Plan Recommendation Using Real-World Public Transportation Information

2nd ACM RecSys Workshop on Recommenders in Tourism

August 27th, 2017
Como, Italy

Victor Anthony Arrascue Ayala          Kemal Cagin Gülsen          Marco Muñiz
Anas Alzogbi                          Michael Färber               Georg Lausen
Tourist trip design problem (TTDP)

- **Input**
  - A user + interests
  - Set of POIs (attractions) + attributes
  - Set of constraints: user’s time budget, opening/closing hours, travel times, etc.

- **Goal:** generate a visit plan (ordered visits of POIs)
  - Real-time requirement
TTDP – example

Time budget: 5h
Start: 09:00
End: 14:00

POI1
O: 09:00
C: 16:00
P: 0.9
V: 4h

POI2
O: 10:00
C: 12:00
P: 0.7
V: 2h

POI3
15 min

15 min

1h

Example diagram showing a touristic plan recommendation using real-world public transportation information.
TTDP – top 2 solutions

**Time budget:** 5h

**Start:** 09:00

**End:** 14:00

<table>
<thead>
<tr>
<th>POI</th>
<th>Time</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>POI1</td>
<td>10:00</td>
<td>0.7</td>
</tr>
<tr>
<td>POI2</td>
<td>11:00</td>
<td>1.0</td>
</tr>
<tr>
<td>POI3</td>
<td>09:00</td>
<td>0.9</td>
</tr>
</tbody>
</table>
TTDP – time dependency constraint

**Time budget:** 5h
**Start:** 09:00
**End:** 14:00

**Profit:** 0.9

POI₁

**Departures:** 11:30 – 1h
12:30 – 1h

O: 11:00
C: 20:00
P: 0.3
V: 1h

POI₂

O: 10:00
C: 12:00
P: 0.7
V: 2h

POI₃

O: 09:00
C: 16:00
P: 0.9
V: 4h

---

27.08.17

A Delay-Robust Touristic Plan Recommendation Using Real-World Public Transportation Information
A Delay-Robust Touristic Plan Recommendation Using Real-World Public Transportation Information

**TTDP – time dependency constraint**

**Time budget**: 5h  
**Start**: 09:00  
**End**: 14:00

- **POI₁**: 
  - O: 11:00  
  - C: 20:00  
  - P: 0.3  
  - V: 1h

- **POI₂**: 
  - O: 10:00  
  - C: 12:00  
  - P: 0.7  
  - V: 2h

- **POI₃**:  
  - Departures: 11:30 – 1h  
  - 12:30 – 1h

**Profit**: 0.9
TTDP – time dependency constraint

(A) Departures: 11:55 – 30 min
12:55 – 30 min

(B) Departures: 12:40 – 15 min
13:40 – 15 min

O: 09:00
C: 16:00
P: 0.9
V: 4h

O: 10:00
C: 12:00
P: 0.7
V: 2h

O: 11:00
C: 20:00
P: 0.3
V: 1h
TTDP – delays

(A) Departures:
11:55 – 30 min + 8 min delay
12:55 – 30 min

(B) Departures:
12:40 – 15 min
13:40 – 15 min

POI₁
O: 09:00
C: 16:00
P: 0.9
V: 4h

POI₂
O: 10:00
C: 12:00
P: 0.7
V: 2h

POI₃
O: 11:00
C: 20:00
P: 0.3
V: 1h
TTDP – delays

(A) Departures:
- 11:55 – 30 min + 8 min delay
- 12:55 – 30 min

(B) Departures:
- 12:40 – 15 min
- 13:40 – 15 min

O: 09:00
C: 16:00
P: 0.9
V: 4h

O: 10:00
C: 12:00
P: 0.7
V: 2h

O: 11:00
C: 20:00
P: 0.3
V: 1h
Route planning for TTDP (advances)

- Transfer Patterns precomputation with Hub Stations, Fast direct-connection queries\(^1\)
- Realistic: traffic, walking between stations, queries between geographic locations instead stations, etc.
- Real-time response (milliseconds)

\(^1\) Hannah Bast et al. Fast Routing in Very Large Public Transportation Networks using Transfer Patterns.
TD(T)OPTW to model TTDP

- OP: Orienteering problem (Knapsack Problem and Traveling Salesman problem)
- TD: time-dependency
- (T): team (plans for multiple days)
- TW: predefined time windows
- Integer linear programming
- OP, OPTW, TDOP, TOP are NP-hard
Iterated Local Search (ILS\textsuperscript{2})

- **Insert Step**
  - Inserts a POI into the solution

- **Shake Step**
  - Removes a POI to escape from local optima

- Generates good solutions (deviates 1.8% from optimal and takes \( \sim 1 \) sec)

\textsuperscript{2} Pieter Vansteenwegen et al. *Iterated local search for the team orienteering problem with time windows.*
Iterated Local Search (ILS²)

- Solution stable for 150 iterations

```
S ← 1;
R ← 1;
NumberOfTimesNoImprovement ← 0;
while NumberOfTimesNoImprovement ≤ 150 do
    while not local optimum do
        Insert;
        if Solution better than BestFound then
            BestFound ← Solution;
            R ← 1;
            NumberOfTimesNoImprovement ← 0;
        else
            NumberOfTimesNoImprovement ← NumberOfTimesNoImprovement + 1;
            Shake Solution (R, S);
            S ← S + R;
            R ← R + 1;
        end if
    end while
    if S >= Size of smallest Tour then
        S ← S - Size of smallest Tour;
    end if
    if R = n / (3^m) then
        R ← 1;
    end if
end while
Return BestFound;
```

² Pieter Vansteenwegen et al. Iterated local search for the team orienteering problem with time windows.
Iterated Local Search (ILS²)

- Solution stable for 150 iterations
- Insert until local optimum

\[
\begin{align*}
S &\leftarrow 1; \\
R &\leftarrow 1; \\
\text{NumberOfTimesNoImprovement} &\leftarrow 0; \\
\text{while} \text{ NumberOfTimesNoImprovement } < 150 \text{ do} \\
&\quad \text{while not local optimum do} \\
&\quad \quad \text{Insert;} \\
&\quad \quad \text{If Solution better than BestFound then} \\
&\quad \quad \quad \text{BestFound } \leftarrow \text{Solution}; \\
&\quad \quad \quad R \leftarrow 1; \\
&\quad \quad \quad \text{NumberOfTimesNoImprovement } \leftarrow 0; \\
&\quad \quad \text{Else} \\
&\quad \quad \quad \text{NumberOfTimesNoImprovement} \\
&\quad \quad \quad \leftarrow \text{NumberOfTimesNoImprovement } + 1; \\
&\quad \quad \text{Shake Solution } (R, S); \\
&\quad S \leftarrow S + R; \\
&\quad R \leftarrow R + 1; \\
&\quad \text{If } S >= \text{Size of smallest Tour then} \\
&\quad \quad S \leftarrow S - \text{Size of smallest Tour}; \\
&\quad \text{If } R == n/(3*m) \text{ then} \\
&\quad \quad R \leftarrow 1; \\
\text{Return BestFound;}
\end{align*}
\]

² Pieter Vansteenwegen et al. Iterated local search for the team orienteering problem with time windows.
Iterated Local Search (ILS$^2$)

- Solution stable for 150 iterations
- Insert until local optimum
- Shake to escape from local optimum

---

2 $^{2}$ Pieter Vansteenwegen et al. Iterated local search for the team orienteering problem with time windows.
Iterated Local Search (ILS$^2$)

- Solution stable for 150 iterations
- Insert until local optimum
- Shake to escape for local optimum
- Variables control number of removed POIs and start position of removal

2 Pieter Vansteenwegen et al. Iterated local search for the team orienteering problem with time windows.
ILS Insert step and route planning
ILS Insert step and route planning
ILS Insert step and route planning

- How to make use of information of the route planner?
- Without compromising the quality of solution
- Without violating the real-time requirement
Current solutions

- Time-independent approximation (e.g. avg. travel times) => infeasible plans
- Pre-compute trip plans between all pairs of POIs and times => not possible in large networks
- Pre-compute travel times exploiting regularities in the schedules => not always regular
- Sacrifice route planning aspects (e.g. multi-modality, transfers, walking, etc.) => unrealistic
Our approaches

- Based on average travel times
- Aligned from time to time with Route Planner information

- Strict ILS (SILS)
- Time-relaxed ILS (TRILS)
- Precise Hybrid ILS (PHILS)
A Delay-Robust Touristic Plan Recommendation Using Real-World Public Transportation Information
Evaluation – set up

- 75 POIs – Izmir
- Public bus transportation (ESHOT) –
  - 7.7K stations and ~300 working bus lines
- 75 x 75 possible start-end location pairs
- Time budget: 4, 6 or 8 hours
- Starting times: 10:00 or 12:00
- 5 different user profiles
- TOTAL: ~170K requests
Evaluation – observations

- AvgILS produces infeasible plans
- RepAvgILS (baseline): simple repair strategy

- More infeasible plans are produced:
  - In the nights (close to end of time window)
  - For large time budgets
Evaluation – observations

- Average travel times are a good estimator!

- 170K requests – only 1.6 K (no delays), 1.0 K (delays) were infeasible
Evaluation – overall score

- Baseline: AvgILS and RepAvgILS

- No Delays:
  - SILS (+0.07%) $\rightarrow$ PHILS (+0.06%) $\rightarrow$ TRILS (+0.05%) $\rightarrow$ RepAvgILS

- Delays:
  - PHILS (+0.05%) $\rightarrow$ SILS (+0.04%) $\rightarrow$ TRILS (+0.035%) $\rightarrow$ RepAvgILS
Evaluation – profit for infeasible plans

- No Delays: SILS (+6.76%) $\rightarrow$ PHILS (+6.06%) $\rightarrow$ TRILS (+5.12%) wrt. to RepAvgILS
- Delays: PHILS (+6.9%) $\rightarrow$ SILS (+6.6%) $\rightarrow$ TRILS (+4.95%) wrt. to RepAvgILS
- SILS and TRILS: recover score of infeasible plans
- PHILS produces alternative solutions for feasible plans
Evaluation – execution times

- Execution times wrt. RepAvgILS:
  - SILS (1.2% slower) $\rightarrow$ TRILS (1.5% slower) $\rightarrow$ PHILS (26.12% slower) wrt. to RepAvgILS

- All < 15 ms x request (on average)
Evaluation – observations

- Delays
  - Not always a bad thing if a system is aware of them
  - New possibilities for traveling might become available!
Conclusions & Future work

- Focus: modeling a realistic scenario
  - Considering delays was possible!
- SILS and PHILS performed the best
- Real-time requirement fulfilled

- Understand better user needs
Thank you!

Any questions?