A simulated annealing based algorithm to generate driving cycle

Yuan Sun

Advisor: Dr. Xu

University of Nevada, Reno
Center of Advanced Transportation Education and Research
Intelligent Transportation System Lab

April 7, 2016
Contents

• Introduction of enhanced prediction of vehicle fuel economy project.
• Existed driving cycle development method.
• Sliding window based method.
• Simulated annealing based method.
• Driving cycle development results.
• Discussions and future researches.
Background: Introduction of enhanced prediction of vehicle fuel economy project

Figure 1. Example of speed profile and fuel consumption measured on paved surface for a diesel truck
Background: Speed – Acceleration Frequency Distribution (SAFD)

![Example SAFD 3D-surface chart](image)

Figure 2. Example SAFD 3D-surface chart
Existed driving cycle development method

- Segmentation-based construction
- Pattern recognition-based construction
- Modal cycle-based construction
Segmentation-based construction

Figure 3. Typical urban and freeway driving cycle

From: http://www.car-engineer.com/fr/les-differents-cycles-de-conduite/#prettyPhoto
Pattern recognition-based construction

Figure 4. Pattern recognition clustering driving cycle example
Modal cycle-based construction

Figure 5. Model cycle based driving cycle cluster example
Compare between Pattern recognition based method and Model cycle-based method

**PR**
- Define kinematic segments first, then clustering from data
- Three trip type: Urban, Rural, Freeway
- Consider main characteristic, share of segment, and transition probabilities

**MC**
- Clustering to model events first, then clustering from model events
- One trip type: Freeway
- Markov Chain, mainly consider transition probabilities
Why we need to develop new method?

1. Raw data is fundamentally different; not continuous.
2. Data set is extremely large.
3. Traffic scenarios type is much more than previous study.
Sliding window method

Figure 7. Sliding window method for driving cycle development.
Sliding window method

Figure 8. Example of the DiffSum value change.
Figure 9. Example of relationship between representative window length and DiffSum.
Sliding window method

Figure 10. LOS D driving cycle based on the sliding window method.
Figure 11. Sliding window limited by the lengths of available trip snippets.
Sliding window method problems

Figure 12. LOS E driving cycle based on the sliding window method (only deceleration period got represented).
Optimization problem forming

- 1. Longest length.
- 2. Smallest Diffsum. (Longer length indicate smaller Diffsum)

Figure 13. Optimization Problem Searching Space
Optimization algorithms

- Global Optimum algorithms
  - Convex Optimization

- Heuristic optimization algorithms
  - Simulated Annealing (SA)
  - Genetic Algorithms (GA)

- Randomized algorithms
Simulated Annealing (SA) Basic Idea

1. First, generate a random solution
2. Calculate its cost using some cost function you've defined
3. Generate a random neighboring solution
4. Calculate the new solution's cost
5. Compare them:
   - If $c_{new} < c_{old}$: move to the new solution
   - If $c_{new} > c_{old}$: maybe move to the new solution
6. Repeat steps 3-5 above until an acceptable solution is found or you reach some maximum number of iterations.
Simulated Annealing (SA) Parameter

- The equation typically used for the acceptance probability is:

\[ a = \exp \left( \frac{c_{\text{old}} - c_{\text{new}}}{T} \right) \]

- This equation means that the acceptance probability:
  - 1) is always > 1 when the new solution is better (has a lower cost) than the old one. Since you can't have a probability greater than 100%, we use \( \alpha = 1 \) in this case.
  - 2) gets smaller as the new solution gets more worse than the old one.
  - 3) gets smaller as the temperature decreases (if the new solution is worse than the old one).
Simulated Annealing (SA) Example

Figure 14. Simulated Annealing Searching Example
A simulated annealing based driving cycle construction method

- **Speed-Acceleration Status Transition Matrix**

<table>
<thead>
<tr>
<th>Speed (MPH)</th>
<th>Acceleration (ft/sec²)</th>
<th>0</th>
<th>…</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15</td>
<td>-15.5</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>-14</td>
<td>-14.5</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>14</td>
<td>14.5</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Figure 15. Example of a SAST matrix
A simulated annealing based driving cycle construction method

Figure 16. Flow chart of SA based algorithm
Driving Cycle compares between SA and sliding window

Figure 17. Driving Cycle compare between SA and Sliding window example
Discussion on SA algorithm

1. The optimization search space can be tremendous. Although there are 11,041 cells in each row of the transition matrix, only a few of them have non-zero probability values. Therefore, in each iteration of the SA algorithm, the next speed-acceleration status is randomly selected second-by-second by using the non-zero probabilities.

2. The total number of each speed-acceleration combination is stored in the SA method. When a speed-acceleration combination is randomly selected as the next status, the total number of this speed-acceleration combination is reduced by 1. When the available number of one speed-acceleration combination is 0, this status will not be selected again.

3. The new solution is by randomly selecting a existed point of the previous driving cycle, then reconstruct all the rest point from this selected point.
Possible alternative method

- Convex Optimization.
- Generic Algorithm.

Figure 18. Convex set solution space (left) and GA work flow (right)
Driving cycle application on ITS

- Hybrid Electrical Vehicle engine mode optimization.

Figure 19. Typical Hybrid Electrical Vehicle engine design
References


QUESTIONS?