Chair of Traffic Engineering and Control
TUM School of Engineering and Design
Technical University of Munich

Poster Number 1084

Automated conflict region calculation for intersection control with automated and connected vehicles

Natalie Steinmetz, Tanja Niels and Klaus Bogenberger

Technical University of Munich, Department of Mobility Systems Engineering, Chair of Traffic Engineering and Control

Motivation / Background

- In scenarios with 100% connected automated vehicles (CAVs), traffic signals could be replaced by autonomous intersection management (AIM) working with V2X communication.
- With AIM, CAVs are assigned arrival times for the intersection, while adjusting their speed individually to arrive without stopping, but at the assigned time. This increases throughput at intersections and reduces vehicle delay (see Figure 1).
- Pedestrians and cyclists can also be integrated into AIM [1], even with greater traffic safety than in today's traffic, by completely blocking all or part of the intersection.
- AIM has evolved over time from First-come, First-served (FCFS) priority policies to optimization-based approaches (see [2] for an overview) that reserve intersection space based on conflict-points [1] instead of tiles at the inner intersection space (see Figure 2).

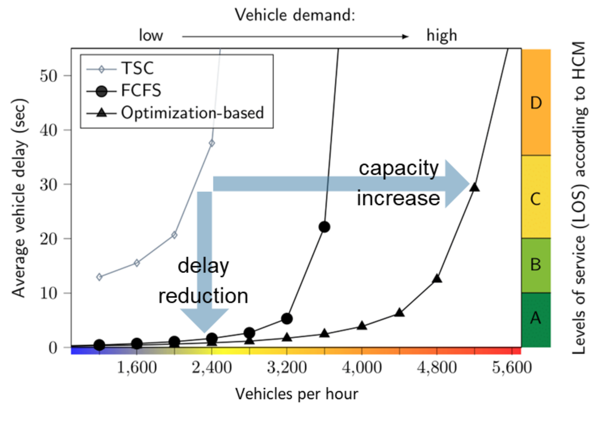

Figure 1: Comparison of vehicle delay at a 4-leg intersection (1 lane per driving direction and leg)

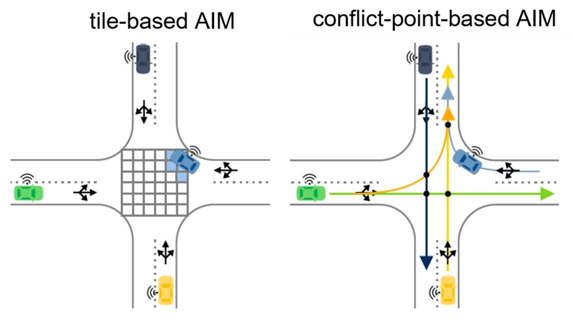

Figure 2: Tile-based versus conflict-point-based intersection management

In a Nutshell:

- For conflict-point-based AIM, conflicting movements need to be recognized and resolved in optimization by shifting vehicle arrival times.
- Conflict points must be calculated and measured for all road users and for all OD relations at the intersection, similar to the calculation of intergreen times in traffic signal control [3].
- Intersection angles must be determined to reserve conflict regions with regard to the road users' dimensions
- With several hundred conflict points per intersection, this is not feasible without automation.
- We implemented a code for automatic intersection point/region calculation in SUMO that is accessible on github.

General Approach

Before a new network/intersection can be simulated under AIM control, the automatic conflict point/region calculation must be carried out once.

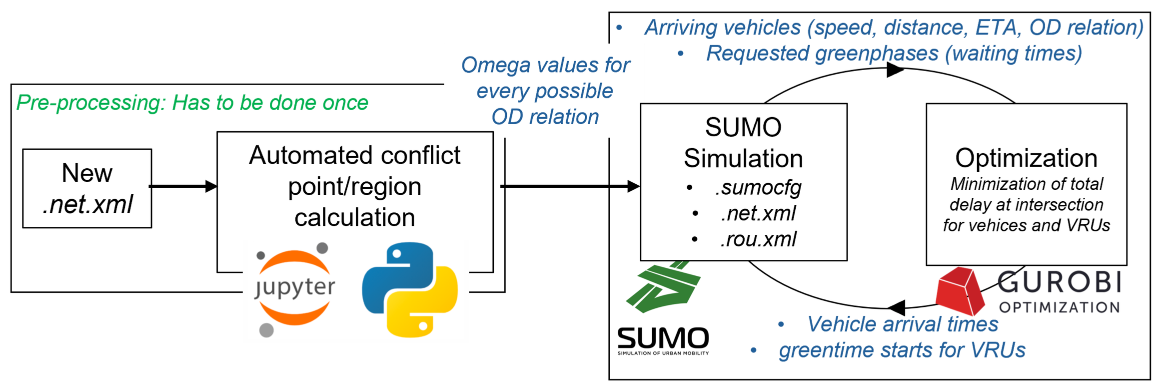

Figure 3: General approach for the simulation of AIM

Automated conflict point/region calculation

.net.xml

Step 1: **Network Topology**

Extract edges and junctions.

for every intersection

Step 2: **Intersection Topology**

Extract connections by searching for internal edges whose ids contain the junction id and get their shape.

```
<edge id=":J0_39" function="internal">
      <lane id=":J0_39_0" index="0" disallow="pedestrian"  speed="6.76“
        length="8.43" shape="608.72,724.50 608.73,723.22 607.88,721.17 "/>
</edge>
```

In addition, search for crossing edges and get their shape.

```
<edge id=":J0_c0" function="crossing" crossingEdges="E0 -E0">
      <lane id=":J0_c0_0" index="0" allow="pedestrian" speed="1.00“
       length="14.80" width="4.00" shape="625.11,709.84 622.94,724.49"/>
</edge>
```

Step 3: **Connect separated connections**

Check if connection is incomplete and find other sections to be connected to.

```
<connection from=":J0_24" to="-E2" fromLane="0" toLane="2" via=":J0_39_0" dir="r" state="m"/>
```

Connect the shapes of incomplete connections and create *shapely Linestring* [4] to enable geometric operations.

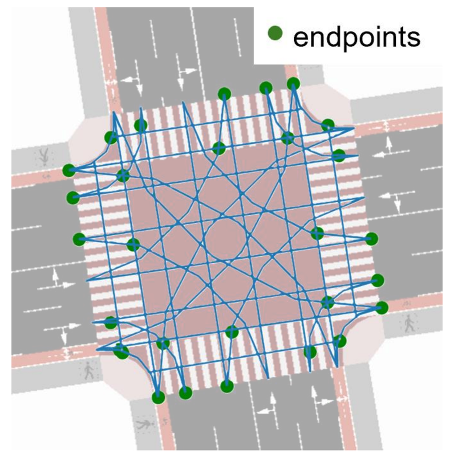

Figure 4: Incomplete connections with endpoints in-between

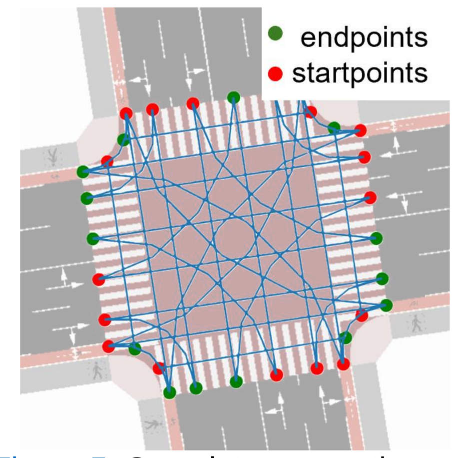

Figure 5: Complete connections with start- and endpoints

Step 4: **Calculate conflict points**

LineString.intersection [5] returns different types of shared geometry (intersection of startpoints, endpoints and “true” intersection points).
Keep only **“true” intersection points** (*LineString.touches* = False [5]) and **intersecting endpoints** (*LineString.touches* = True).

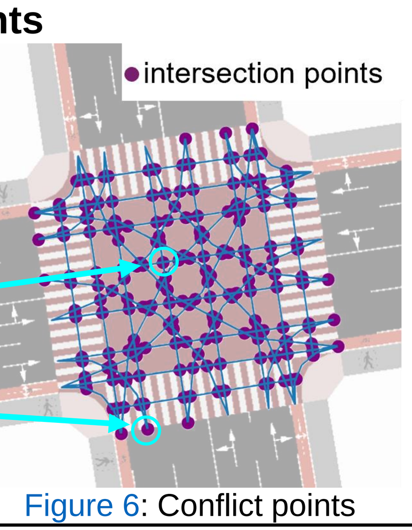

Figure 6: Conflict points

Step 5: **Calculate conflict regions**

The size of a conflict region depends on the angle at which the connections intersect. The angle is calculated by using the *numpy.dot* [6] product. The more acute the cutting angle, the larger the conflict region along the connection. For each conflict pair (a, b) the distance to the beginning of the conflict region, denoted by ω_{1i}, and the distance within the conflict region, denoted by ω_{2i}, is calculated (i in $\{a, b\}$) (see Figure 7).

$$omegaValues[OD_a, OD_b] = [\omega_{1a}, \omega_{1b}, \omega_{2a}, \omega_{2b}]$$

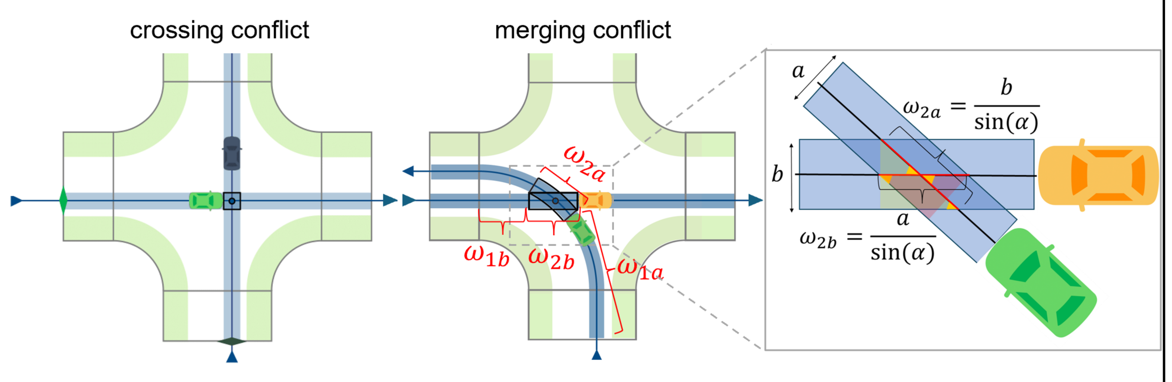

Figure 7: Calculating the dimension of the conflict region dependent on the intersecting angle

For pedestrians, the whole crossing is blocked ($\omega_1 = 0$, $\omega_2 = width_{road}$).

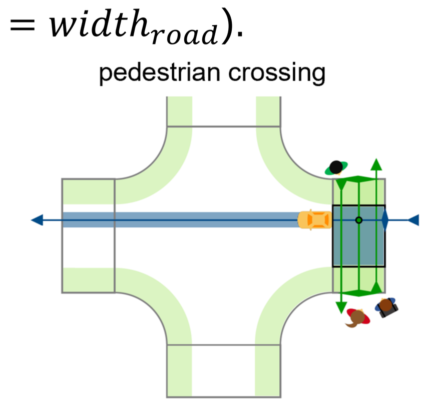

Figure 8: Blocking of pedestrian crossings

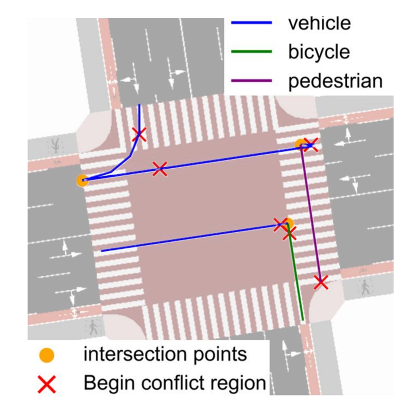

Figure 9: Some examples for calculated conflict regions

Further applications

Determine connection/OD-related conflict numbers and use them for routing and global control, e.g. by creating routes with many right turns (see Figure 11).

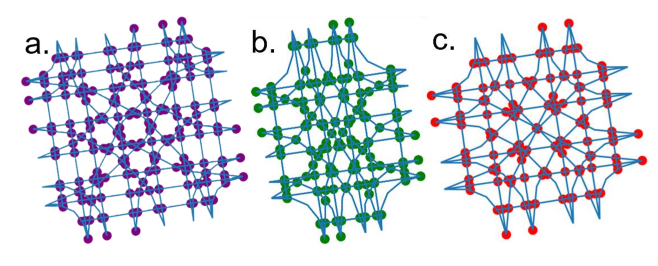

Figure 10: Number of conflict points for different intersection layout with pedestrians and separate bicycle tracks (a: 416, b: 364, c: 320)

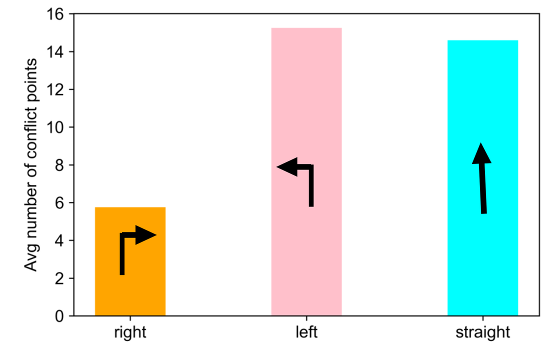

Figure 11: Average number of conflict points for different turning directions, considering different intersection layouts

References

[1] Niels, T., Bogenberger, K., Papageorgiou, M., & Papamichail, I. (2024). Optimization-Based Intersection Control for Connected Automated Vehicles and Pedestrians. Transportation Research Record, 2678(2), 135-152.
[2] Zhong, Z., Nejad, M., & Lee, E. E. (2021). Autonomous and Semiautonomous Intersection Management: A Survey. IEEE Intelligent Transportation Systems Magazine, 13(2), 53–70.
[3] FGSV (2015). Handbuch für die Bemessung von Straßenverkehrsanlagen: HBS. Köln: FGSV Verlag.
[4] https://shapely.readthedocs.io/en/stable/reference/shapely.LineString.html
[5] https://shapely.readthedocs.io/en/stable/manual.html
[6] https://numpy.org/doc/stable/reference/generated/numpy.dot.html

Contact information

Natalie Steinmetz, M.Sc.
Chair of Traffic Engineering and Control
Technical University of Munich
natalie.steinmetz@tum.de

Github

Jupyter notebook for automated conflict point/region calculation is available here:

SUMO User Conference 2024
German Aerospace Center (DLR)
Berlin - May 13-15, 2024