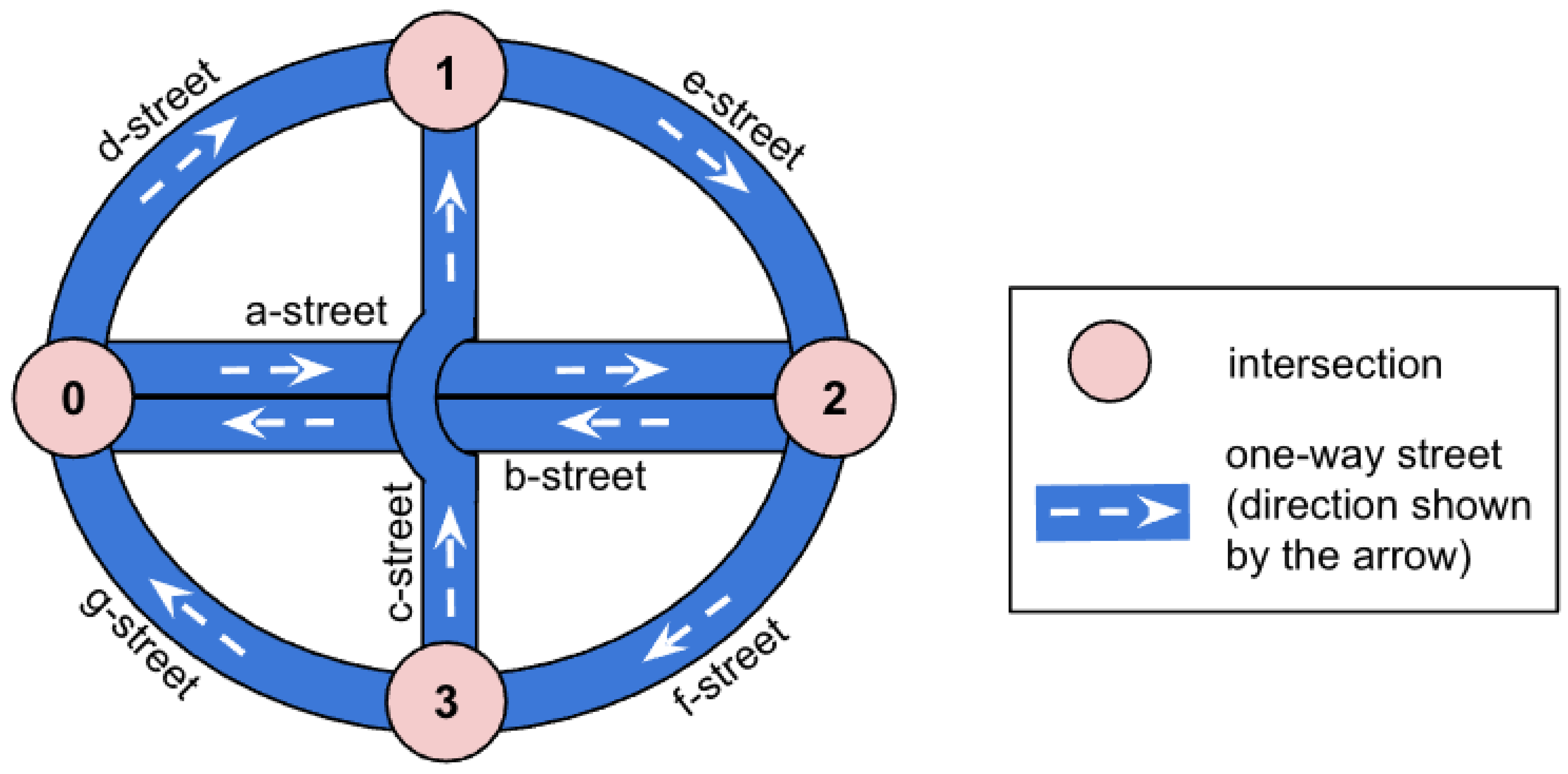
Traffic Signal Optimization

David Ruda
Supervisor: RNDr. Jiří Fink, Ph.D.
2025

Problem

Traffic congestion is a growing issue in cities. Optimizing traffic signals is a cost-effective way to reduce delays and emissions without changing the physical infrastructure. The Google Hash Code 2021 competition introduced a simplified version of this real-world problem:

*Given a city plan describing **intersections**, **streets**, and **cars** with planned paths through the city, optimize **traffic light schedules** to maximize the number of cars reaching their destination before the deadline, while minimizing the overall time spent in traffic.*



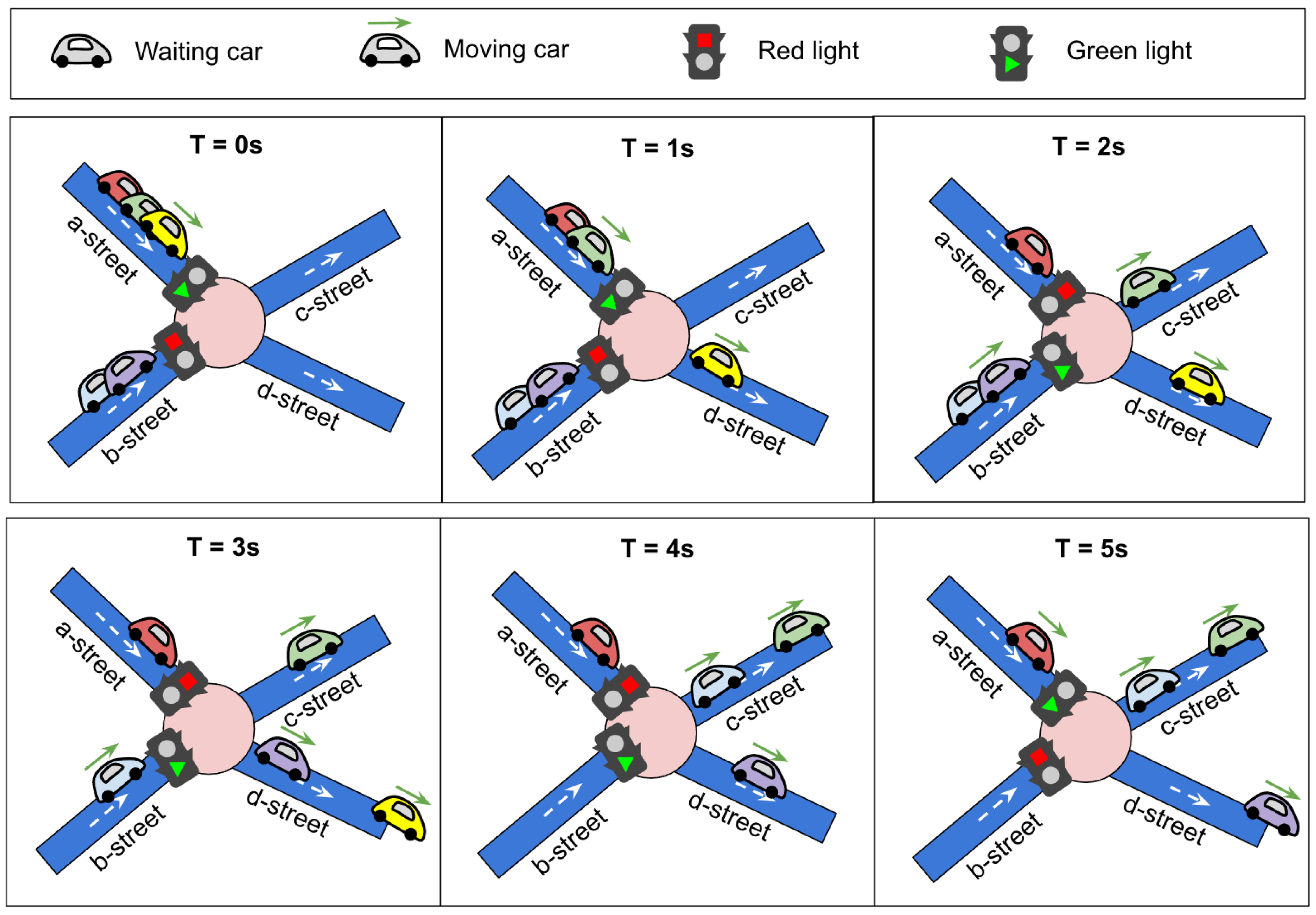
Example of a city plan [Google Hash Code 2021].

Thesis Goals

- Implement a **fast and efficient C++ simulator** for the problem and wrap it as a **Python package** for convenient use
- Integrate the simulator into an optimization pipeline with three heuristic algorithms: **Genetic Algorithm**, **Hill Climbing**, and **Simulated Annealing**
- Compare the performance of these algorithms on provided competition datasets

Traffic Light Schedule for one Intersection

Each incoming street has two parameters: the **order** in which it gets a green light, and the **duration** of the green light.



Example of a schedule [Google Hash Code 2021].
First, a-street has a green light for 2 seconds, then b-street for 3 seconds.
One car can pass each second.

github.com/davidruda/Traffic-Signal-Optimization

Score

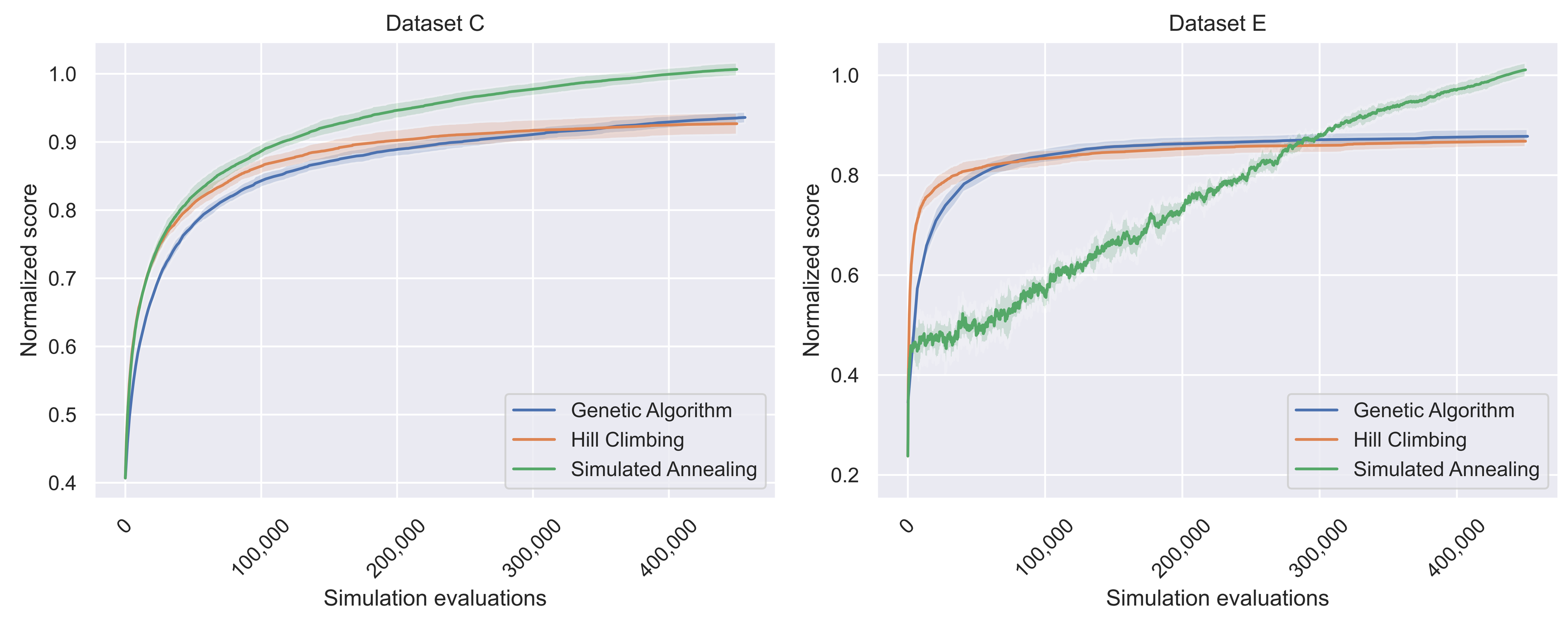
The total score of the schedules $\boldsymbol{\theta}$ is defined as

$$SCORE(\boldsymbol{\theta}) = \sum_{c \in C} \begin{cases} \mathrm{F} + (\mathrm{D} - t(c;\boldsymbol{\theta})), & \text{if } t(c;\boldsymbol{\theta}) \leq \mathrm{D}, \\ 0, & \text{otherwise}, \end{cases}$$

i.e., cars arriving before or at the simulation end get the bonus plus extra points for arriving early; cars arriving later get zero.

D - simulation duration
F - fixed bonus for reaching the destination
$t(c;\boldsymbol{\theta})$ - arrival time of car c

Optimization Process



Convergence of optimization algorithms on datasets C and E.
All algorithms have a fixed budget of simulation evaluations.

Results

Dataset	# Params	Genetic Algorithm	Hill Climbing	Simulated Annealing
B	5,974	0.91	0.87	**0.97**
C	14,008	0.94	0.93	**1.01***
D	167,748	0.90	**0.92**	0.92
E	1,386	0.88	0.87	**1.01***
F	10,002	0.93	0.93	**0.94**

The scores are averaged over 10 seeds and normalized:
0.00 - our baseline, 1.00 - maximum known score
**New maximum known score*

Main Result

Simulated Annealing consistently outperforms the other methods, setting new maximum known scores on datasets C and E.

Conclusions

- All three algorithms achieved good results across all datasets
- Genetic Algorithm's broad search is less beneficial for this particular problem
- Single-state methods like Simulated Annealing seem more suitable

Possible future work:

- Compare methods under equal runtime budget, not only by equal simulation evaluations
- Test other single-state methods (Tabu Search, Iterated Local Search)