
Traffic Optimization

Task: Optimize traffic light timing with reinforcement learning.

Prof. Swati Mali

Kashish Mamania 16010122104

Riddhi Patil 16010122138

Sania Parekh 16010122132

Tanaya Pawar 16010122143

Sarthak Pokale 16010122146

1. Problem Definition and Objectives of Work

Urbanization has led to a dramatic increase in traffic congestion, especially in metropolitan cities. Traditional traffic signal systems operate on fixed schedules, failing to adapt to real-time traffic conditions. This inefficiency results in longer vehicle waiting times, increased fuel consumption, and higher pollution levels. The core problem addressed by this project is the **optimization of traffic signal timings at intersections** to reduce congestion and improve overall traffic flow.

Objectives:

- **Minimize Vehicle Waiting Time:** Reduce the average time vehicles spend idling at signals.
- **Optimize Green Light Durations:** Dynamically adjust signal timings based on simulated traffic density and flow.
- **Implement Genetic Algorithm (GA) Strategy:** Use evolutionary techniques (selection, crossover, mutation) to evolve efficient signal plans.
- **Simulate Realistic Traffic Conditions:** Integrate with the SUMO simulator to test signal plans in a controlled, urban environment.
- **Improve Intersection Efficiency:** Enhance vehicle throughput and reduce congestion at major junctions.
- **Develop a Scalable and Adaptive Model:** Ensure the system can be extended to more complex intersections and potentially real-time scenarios.

2. Datasets Used

- **Synthetic Initial Population:**
The genetic algorithm starts with a synthetic, programmatically generated set of candidate signal timings (e.g., lists like `[50, each representing green light durations for different phases at the intersection. These are not based on real-world data but are created within the code to serve as the initial search space for optimization.
- **SUMO Network and Route Files:**
The simulation relies on SUMO configuration files:
 - **Network file (.net.xml):** Describes the road network and intersection layout.
 - **Route file (.rou.xml):** Specifies vehicle flows and routes.
 - **Simulation config (city.sumocfg):** Integrates the above and defines the simulation scenario.
These files together constitute the dataset for the simulation, modeling the traffic environment and vehicle demands.
- **Simulation Output Data:**
During each run, SUMO generates data such as average waiting time, travel time, number of vehicles arrived/departed, and fuel consumption. This output is used to evaluate the fitness of each candidate signal plan and to analyze results.

3. Tools/Libraries/APIs Used

- **SUMO (Simulation of Urban MObility):**
An open-source, microscopic traffic simulation package used to model and simulate urban traffic networks and collect traffic data.
- **TraCI (Traffic Control Interface):**
A Python API for real-time interaction with SUMO, enabling dynamic control of traffic lights and retrieval of simulation data.
- **Python:**
The main programming language for implementation, utilizing:
 - numpy for numerical operations and randomization in the genetic algorithm.
 - Standard libraries (random, copy, os, sys, subprocess, optparse) for utility functions and process management.
 - Custom modules (junction, device, phaseConfig) for modeling intersections and traffic light logic.
- **SUMO Configuration Files:**
 - .sumocfg (simulation configuration)
 - .net.xml (network)
 - .rou.xml (routes)

These tools collectively enable the simulation, optimization, and analysis of traffic signal timing strategies.

4. Implementation Details

The project employs a **simulation-based optimization approach** using a genetic algorithm integrated with SUMO:

- **Traffic Environment Modeling:**
The intersection and road network are defined in SUMO using XML configuration files.
- **Genetic Algorithm Workflow:**
 1. **Initialization:**
Start with a population of randomly generated signal timing plans (synthetic data).
 2. **Simulation:**
For each individual (signal plan), apply it in the SUMO simulation and collect performance metrics (waiting time, travel time, fuel consumption).
 3. **Fitness Evaluation:**
Calculate a fitness score for each plan based on the collected metrics, aiming to minimize waiting and travel times.
 4. **Selection:**
Choose the best-performing plans for reproduction.
 5. **Crossover and Mutation:**
Generate new plans by combining and randomly altering selected plans.
 6. **Iteration:**
Repeat the process for multiple generations, with each generation expected to yield better signal timings.
- **Integration with SUMO:**
TraCI is used to dynamically set signal phases and retrieve real-time traffic data during each simulation step.
- **Result Logging:**
At each step and generation, relevant metrics are logged for analysis and comparison.

5. Result Analysis

- **Performance Metrics:**
The system tracks phase averages, fitness values, number of arrived and departed cars, average waiting time, travel time, and fuel consumption for each simulation step.
- **Observed Trends:**
 - Early steps show high waiting times and fitness values, indicating suboptimal signal plans.
 - As the genetic algorithm evolves the population, fitness improves, and waiting/travel times decrease.
 - Fuel consumption also trends downward, reflecting smoother traffic flow.

- **Tabulated Results:**

The table below summarizes key metrics across simulation steps (see also the PPT for the full table):

Step	Phase Average	Fitness	Arrived Cars	Departed Cars	Simulation Time	Avg Waiting Time	Avg Travel Time	Avg Fuel Consumption
0	44	2984.64	0	43	44.0	24.84	10.0	1037.38
1	39	8640.92	0	81	83.0	20.15	331.0	907.15
2	36	14159.85	1	100	119.0	18.79	474.33	866.35
3	39	27237.81	6	100	158.0	2425.27	548.0	832.88
4	25	15363.06	8	100	183.0	3869.82	651.20	835.16
5	49	795.94	46	100	232.0	5950.35	848.0	833.33
6	23	308.19	57	100	255.0	6477.72	901.71	828.07
7	20	10.29	59	100	275.0	5669.32	824.63	826.86
8	30	11.52	76	100	305.0	5040.94	741.33	819.57
9	27	10.54	77	100	332.0	4538.16	682.40	810.98
10	34	12.48	82	100	366.0	4126.57	626.18	804.90
11	44	15.18	92	100	410.0	3783.55	577.0	803.28
12	20	6.92	97	100	430.0	3493.28	536.15	802.15
13	19	6.53	98	100	449.0	3244.46	497.86	801.30
14	33	11.20	98	100	482.0	3028.81	464.67	798.75
15	53	17.82	99	100	535.0	2840.11	435.63	799.00

- **Conclusion:**

The genetic algorithm successfully reduces waiting times and improves intersection efficiency over multiple generations. The simulation demonstrates that evolutionary optimization can adapt signal timings to dynamic traffic conditions, outperforming static, fixed-time schedules. The approach is scalable and forms a strong foundation for future real-time and multi-objective traffic management solutions.