CSE-400 SECTION 1

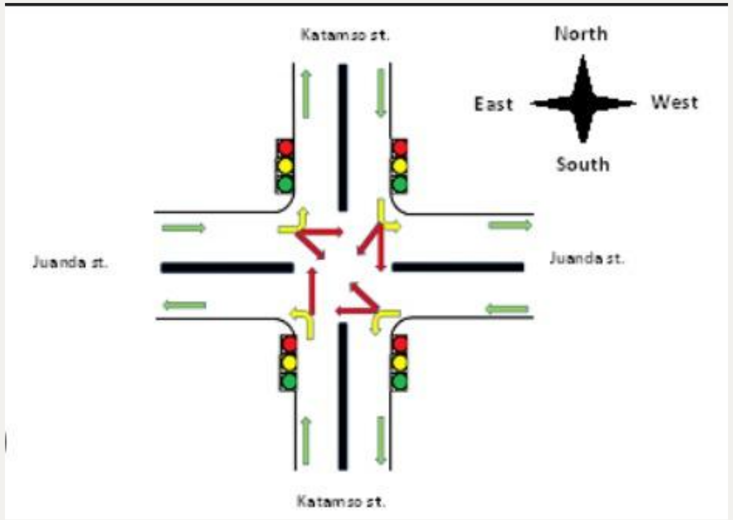
Prof. Dhaval Patel

Traffic Light Optimization for higher traffic flow and reduced waiting time

Group - s1_its_5

-Sakina Jambughodawala (AU2340114)
-Parin Patel (AU2340243)
-Maanya Patel(AU2340087)
-Hevit Makavana (AU2340194)

Problem Formulation



https://images.app.goo.gl/6DPjXQmxPmnSMGvAA

https://images.app.goo.gl/jy14Pv5rJUN444gj6

https://images.app.goo.gl/wDkFuGbDFFTkbctB8

T1: Mathematical Model

Sakina - AU2340114

Random Variables

1. Number of Vehicles, N:
- Discrete
- PMF
- $Ni \leq Nmax$
- Ti (cleared traffic- vehicles passed during green light)
- Ri (residual traffic - vehicles left behind after green light)
- Depends on λi and μ

2. Arrival Rate, λi :
- Continuous
- PDF: Poisson's distribution
- λi = Ni per sec

3. Waiting Time, W:
- Continuous
- PDF: Normal Distribution
- k-gi : W during red light
- $Ri = [(k - gi).\lambda i + gi.\lambda i] - T$: W after green light and which will be cleared in next cycle

4. Traffic Light State, TS:
- Discrete
- PMF
- $TSgi=k(\lambda maxi / \sum\lambda maxi)$

5. Service Rate, μ:
vehicles cleared when light is green

Ahmedabad University

https://files.campuswire.com/e5a84109-702a-42ef-929f-f6d7c7febdee/e36df6c5-4cd7-48ad-9cb0-0e56caf85d12/1-s2.0-S1110016824004344-main.pdf

At an intersection, there will be four paths for vehicles to arrive.
Therefore, total time: $\sum g_i = k$ where k=120sec normally constant distribution=k/4

But we need to dynamically reduce waiting time

Green light time allocation: TSgi

Let λmax be the maximum arrival rate and higher arrival rate gets higher weight gi $\propto$ wi.
So, $w_i = \frac{\lambda_{maxi}}{\sum \lambda_{maxi}}$

Therefore, green light allocation, $g_i = k \cdot w_i$ where $\sum g_i = k$

$$g_i = \frac{k \cdot \lambda_{maxi}}{\sum \lambda_{maxi}}$$

Traffic passed during gi time: Ti

No. of vehicles passed during gi time: μ=20 vehicles per second when k=120 sec

$$T_i = \mu \cdot g_i = \mu \cdot \frac{k \cdot \lambda_{maxi}}{\sum \lambda_{maxi}}$$

Waiting time for new vehicles arriving:

- New vehicles waiting time (per path):

$$k - g_i$$

Traffic Accumulated when light is red:

- Vehicles accumulated during red light:

$$(k - g_i) \cdot \lambda_i$$

- Vehicles passing during green light:

$$g_i \cdot \lambda_i$$

If the green time is not enough, some vehicles remain waiting for next cycle which will be more than k.

- Total accumulated traffic (both red and green periods):

$$(k - g_i) \cdot \lambda_i + g_i \cdot \lambda_i = k \cdot \lambda_i$$

Residual Traffic (Ri)

- Traffic passed during green time:

$$T_i = \mu \cdot g_i$$

- Total accumulated traffic during cycle:

$$(k - g_i)\lambda_i + g_i\lambda_i = k\lambda_i$$

- Residual traffic:

$$R_i = [(k - g_i)\lambda_i + g_i\lambda_i] - T_i$$

- Condition:
  - If Ri=0: All vehicles cleared during green signal
  - If Ri>0: Some vehicles remain waiting for the next cycle

The system can adjust gi (green time) in the next cycle to gradually clear the residual traffic.

T2: Coding

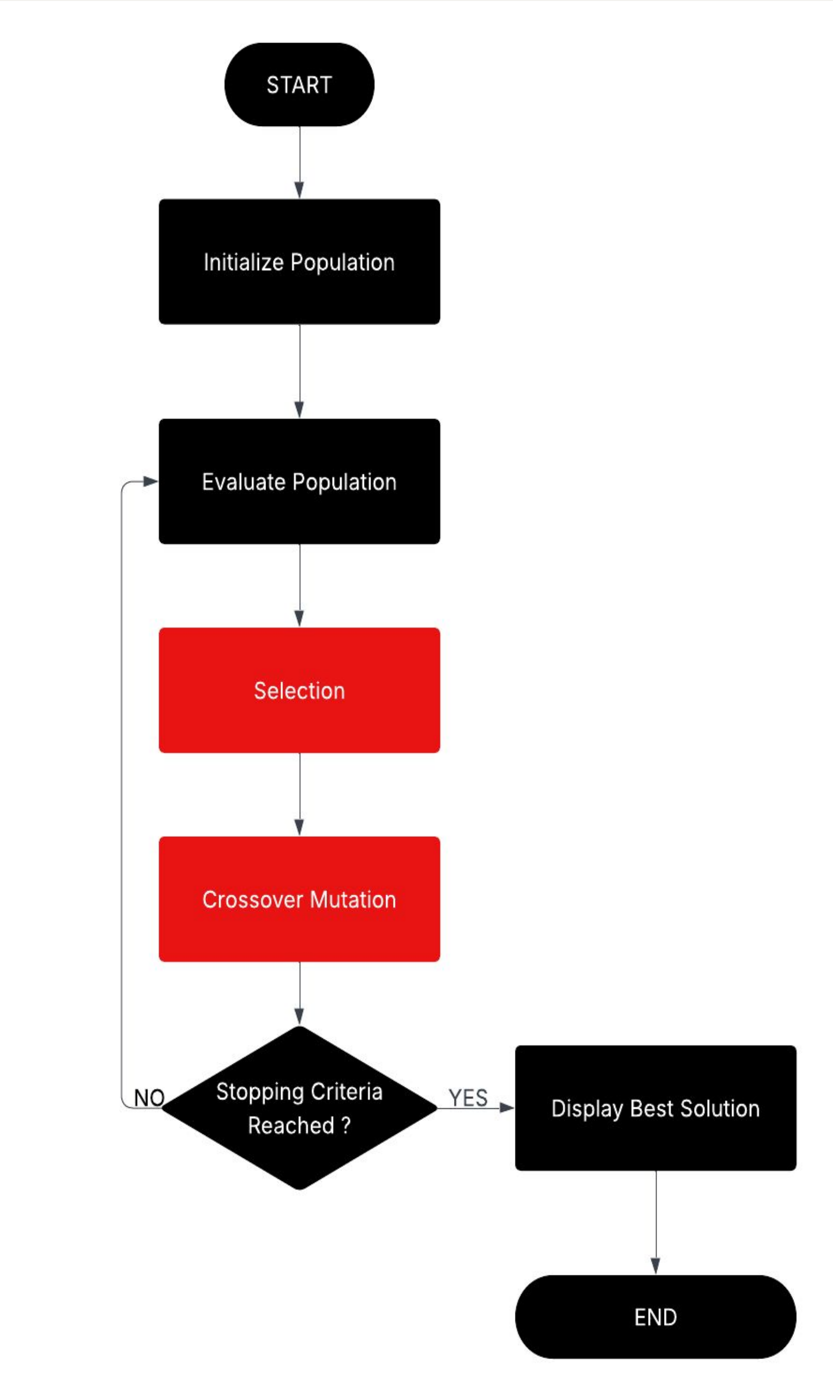
Parin - AU2340243

```
def crossover_parent(chromosome_male,chromosome_female,crossover_rate = crossover_rate):…
```

Combines durations and traffic light states from two parent chromosomes to create a new child solution

Applies random changes to durations and traffic light states to introduce genetic diversity

```
def mutate_chromosome(chromosome,duration_mutation_rate=duration_mutation_rate, …
```

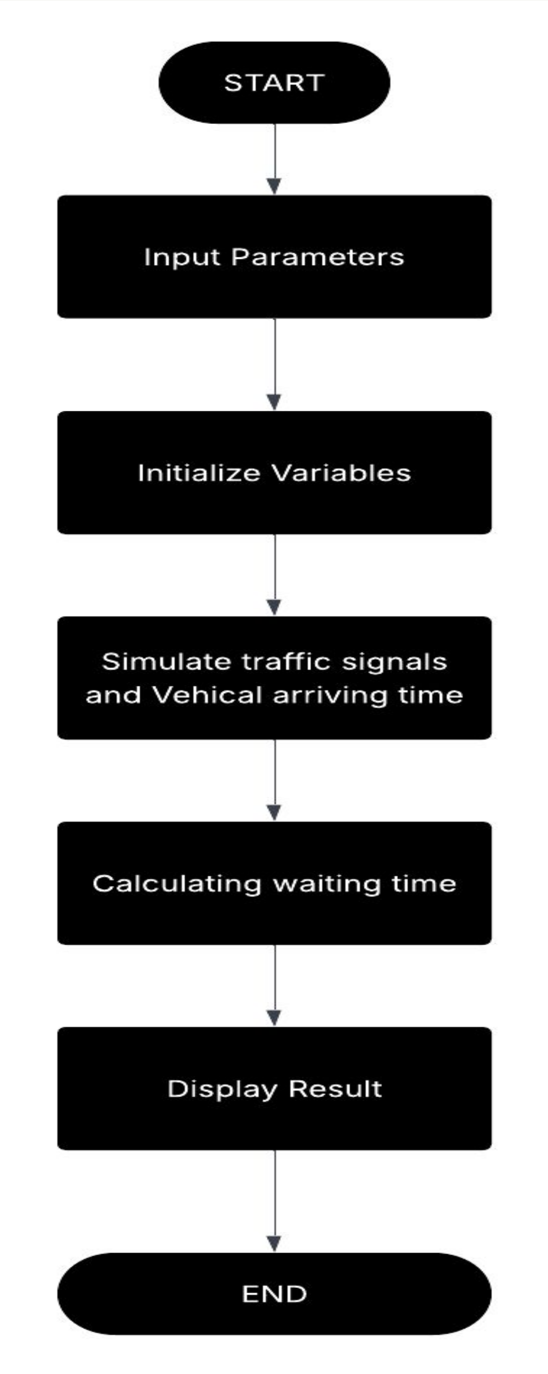


```python
class TrafficSignalSimulation:
    def run_cycle(self, wi):
        ni = [round(self.total_cars * w) for w in wi]
        λ = [round(n / self.k, 4) for n in ni]
        gi = [round(self.k * w, 2) for w in wi]
        Ti = [min(round(self.μ * g, 2), n) for g, n in zip(gi, ni)]
        Wi = [round(self.k - g, 2) for g in gi]
        Ri = [round(n - t, 2) for n, t in zip(ni, Ti)]

        return {
            'λ': λ, 'g': gi, 'T': Ti,
            'W': Wi, 'R': Ri
        }

    def simulate(self):
        for cycle in range(self.num_cycles):
            wi = self.generate_weights()
            result = self.run_cycle(wi)
            self.history.append(result)
```

This code simulates traffic at signals using fixed rules. It calculates how many cars arrive, pass, and wait in each cycle.

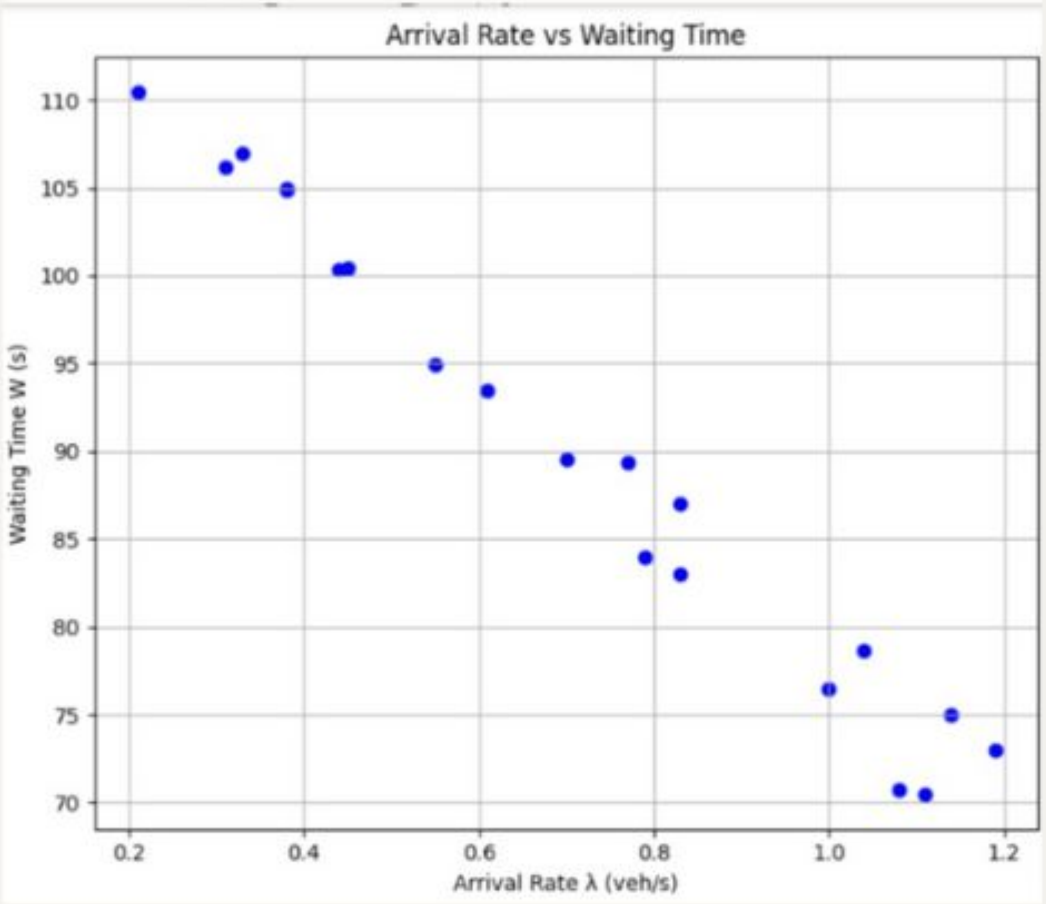


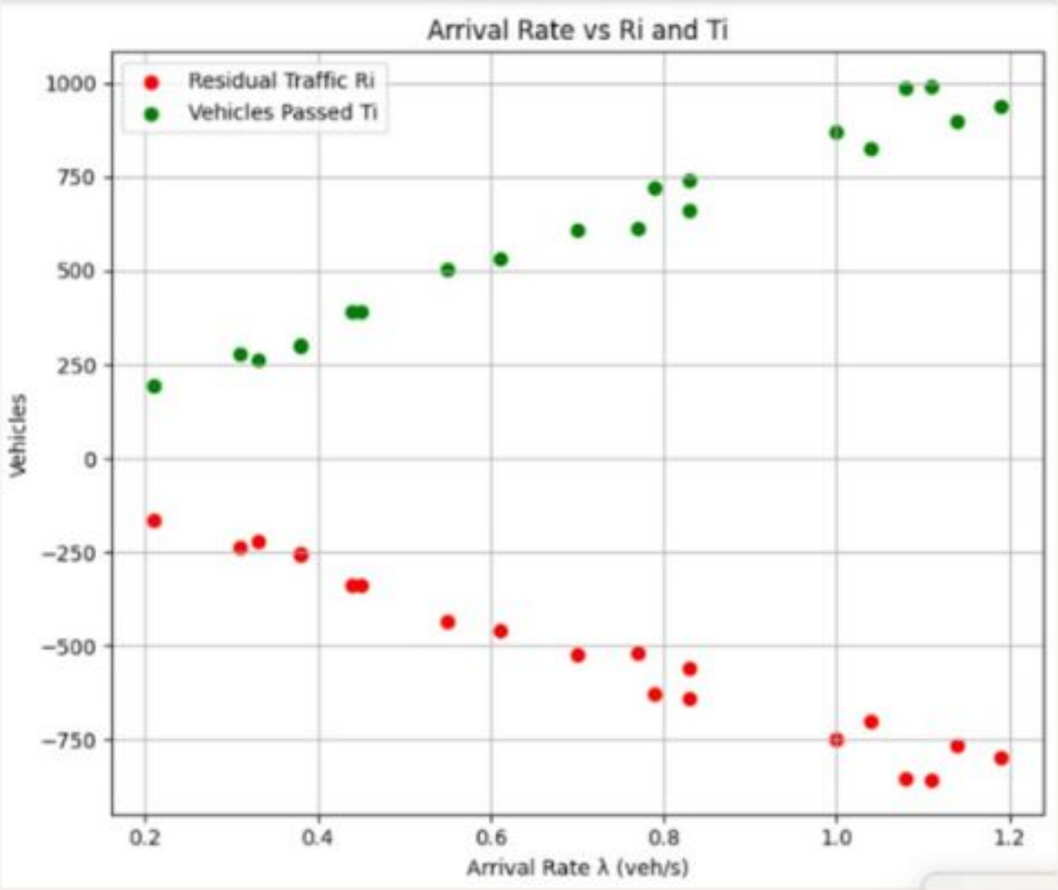
Ahmedabad University

T3: Inferences

Maanya- AU2340087

Graphical Respresntation of Mathematical Model





Ahmedabad University

File Edit Selection View Go Run ···

Traffic-Lights-Genetic-Algorithm-main

Simulation_Generator.py | Genetic_Algorithm.py | main_run.ipynb

Simulation_Generator.py > run_SUMO

```python
def run_SUMO(config_file):
        print("Launching SUMO")
        start([sumoBinary, "-c", config_file,'--start'])
        for step in range(end_time):
            simulationStep()
            time.sleep(0.01)



#Main

#Define the parameters and generate a simulation
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
> python -u "c:\Users\Lenovo-IT\Downloads\Traffic-Lights-Genetic-Algorithm-main\Traffic-Lights-Genetic-Algorithm-main\Simulation_Generator.py"
Success.
```

EXPLORER

TRAFFIC-LIGHTS-GENETIC-ALGORITHM-MAIN

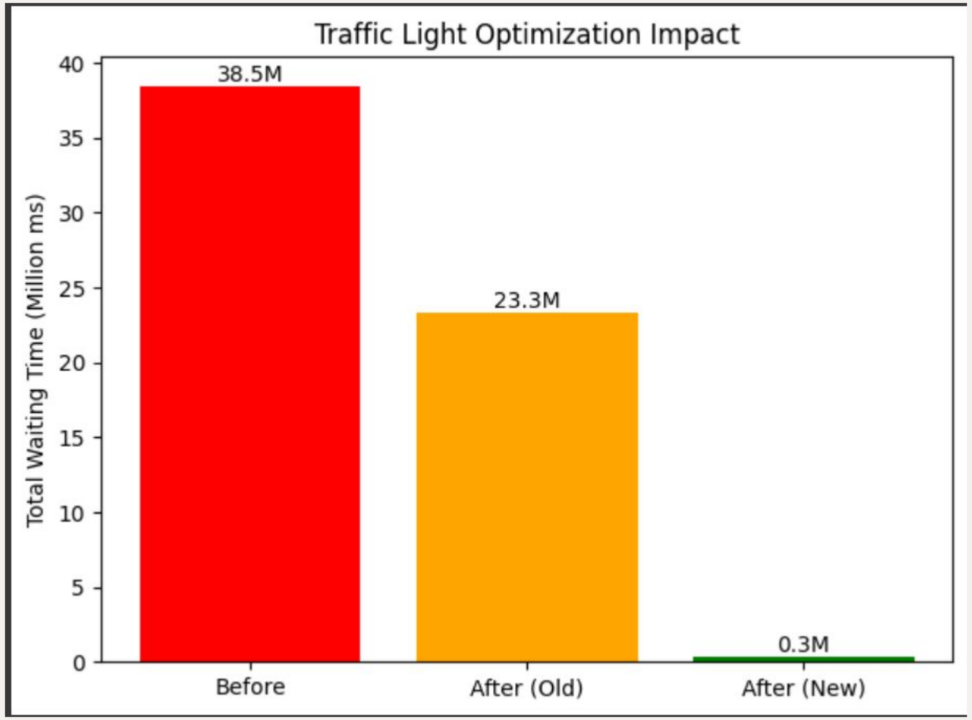
- __pycache__
- wandb
- .gitattributes
- 78f7b89ecaf707478bbad723872325f513b...
- Genetic_Algorithm.py
- main_run.ipynb
- Manhattan5x3.net.xml
- Manhattan5x3.rou.alt.xml
- Manhattan5x3.rou.xml
- Manhattan5x3.sumocfg
- Project Presentation.pptx
- README.md
- Simulation_Generator.py
- tempCodeRunnerFile.py
- trips.trips.xml

OUTLINE

TIMELINE

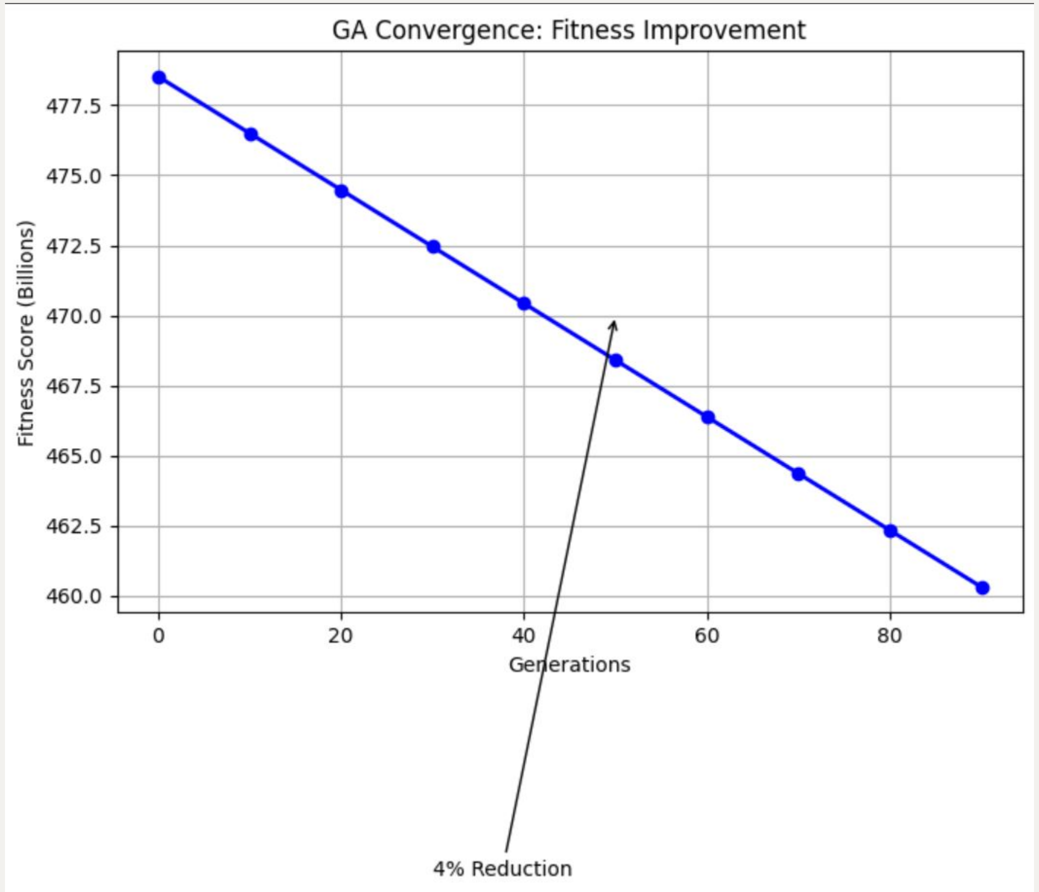
Run Testcases ⊗ 0 ⚠ 0 Ln 110, Col 27 Spaces: 4 UTF-8 LF Python Signed out 3.12.1 Go Live

CS Perspective



Waiting Time Reduction:

- Raw: 38,485,553 ms → 341,071 ms (99% decrease, verify vehicle count).
- Per Vehicle: ~120 sec → ~3.4 sec (likely sparse traffic).

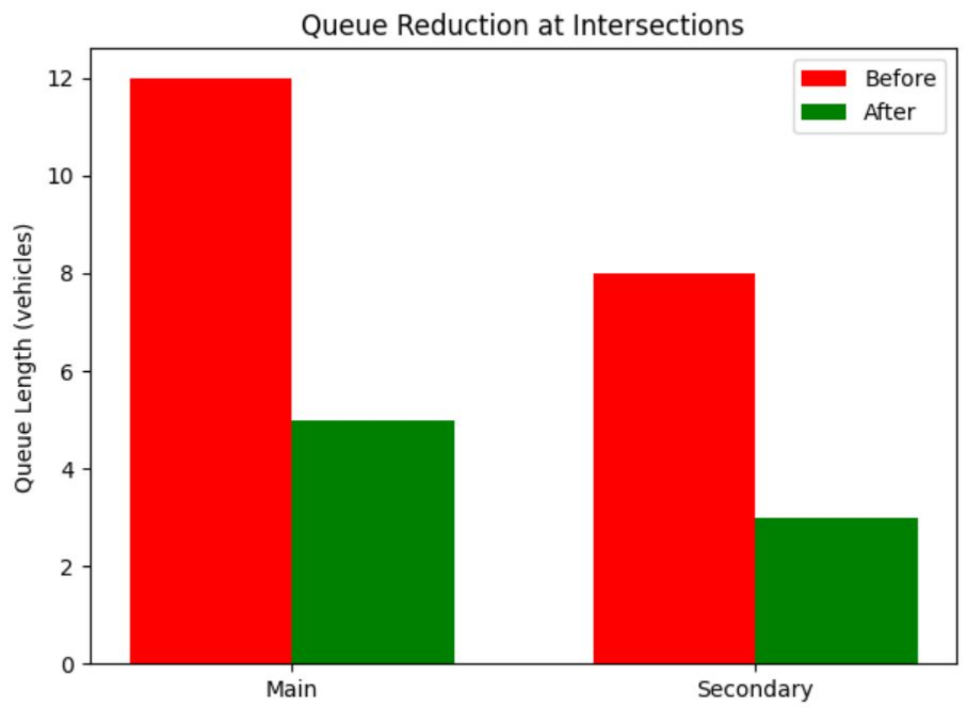


Fitness Score Improvement:

- Initial: 478,518,939,095 → Optimized: 460,310,502,802 (~4% reduction)
- Indicates successful convergence of the GA.

Ahmedabad University

Domain Perspective



Optimization Results Summary

	Metric	Before	After	Improvement
0	Waiting Time	120 sec/veh	3.4 sec/veh	97%
1	CO_2 Emissions	904M mg	500M mg	45%
2	Queue Length	12 vehicles	5 vehicles	58%

- **Queue Lengths:**

Main Intersection: 12 → 5 vehicles (58% shorter queues).

Secondary Intersection: 8 → 3 vehicles (62% improvement).

- **Vehicle Throughput**
- **Safety**

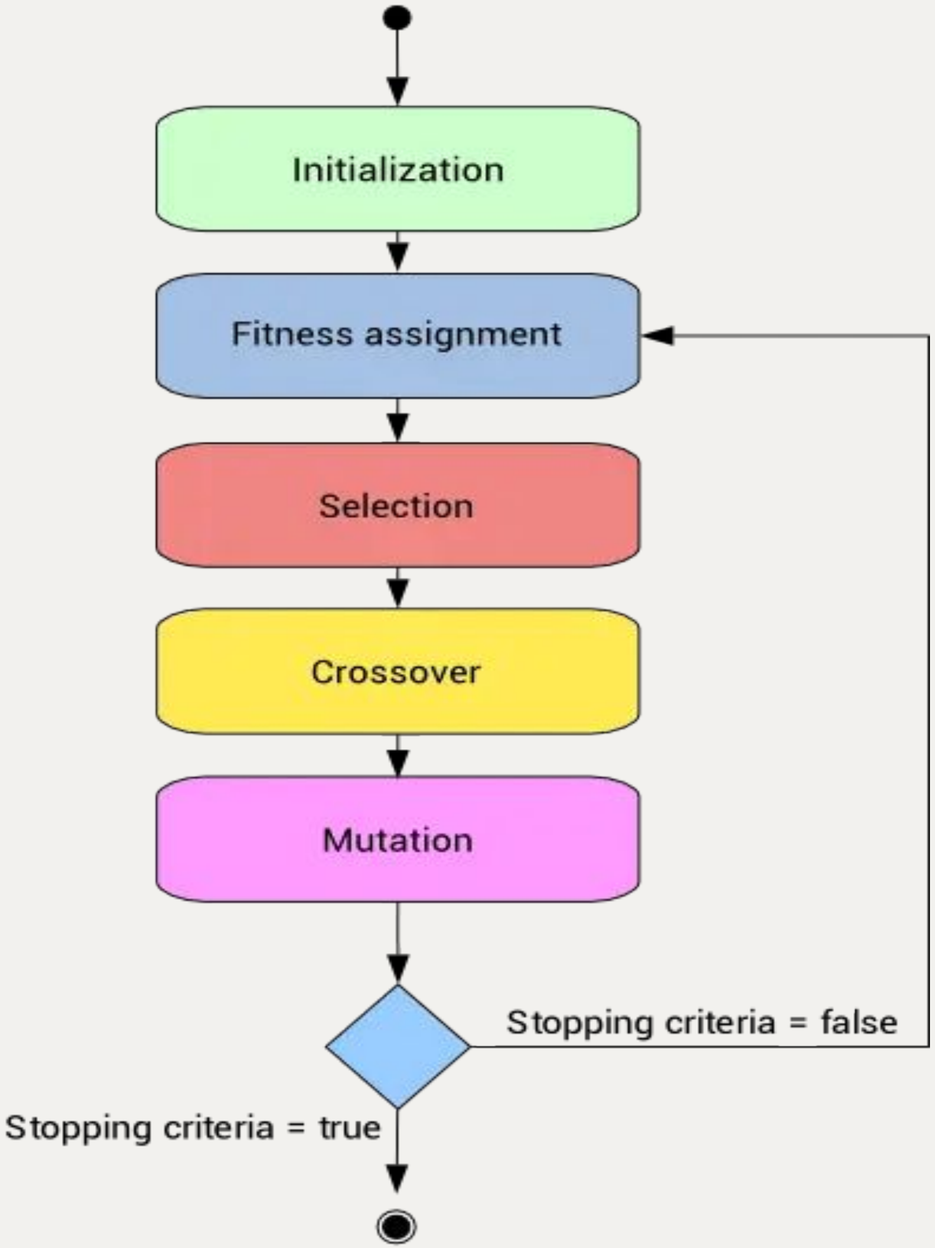
T4: Randomized Algorithm

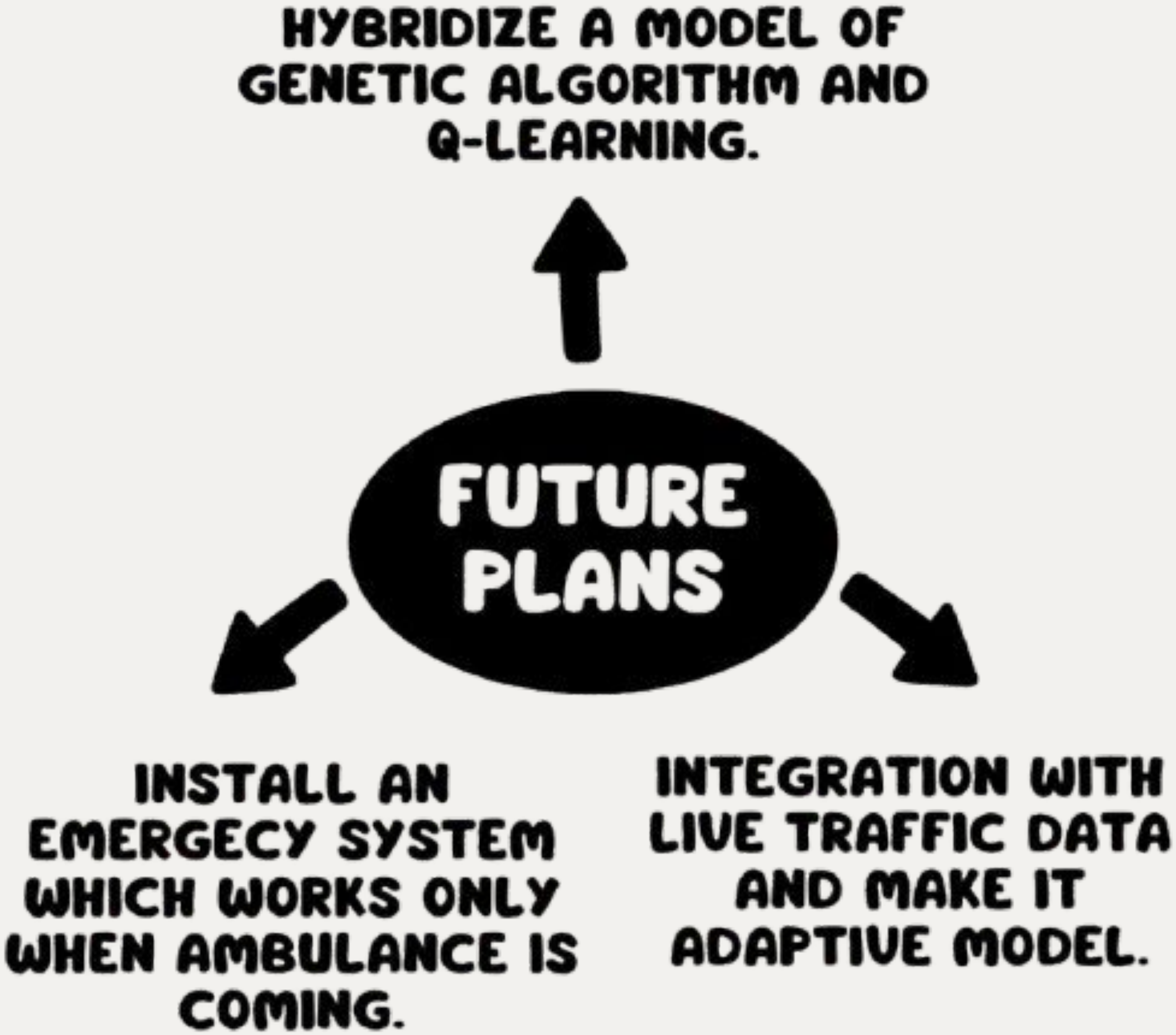
Hevit - AU2340194

Feature	Genetic Algorithm (GA)	Q-Learning
Approach	Evolution-based	Trial & error learning
Adaptability	Slow, needs multiple runs	Fast, adapts in real-time
Decision Making	Best solution from population	Learns best action per state
Best For	Offline optimization	Real-time traffic control

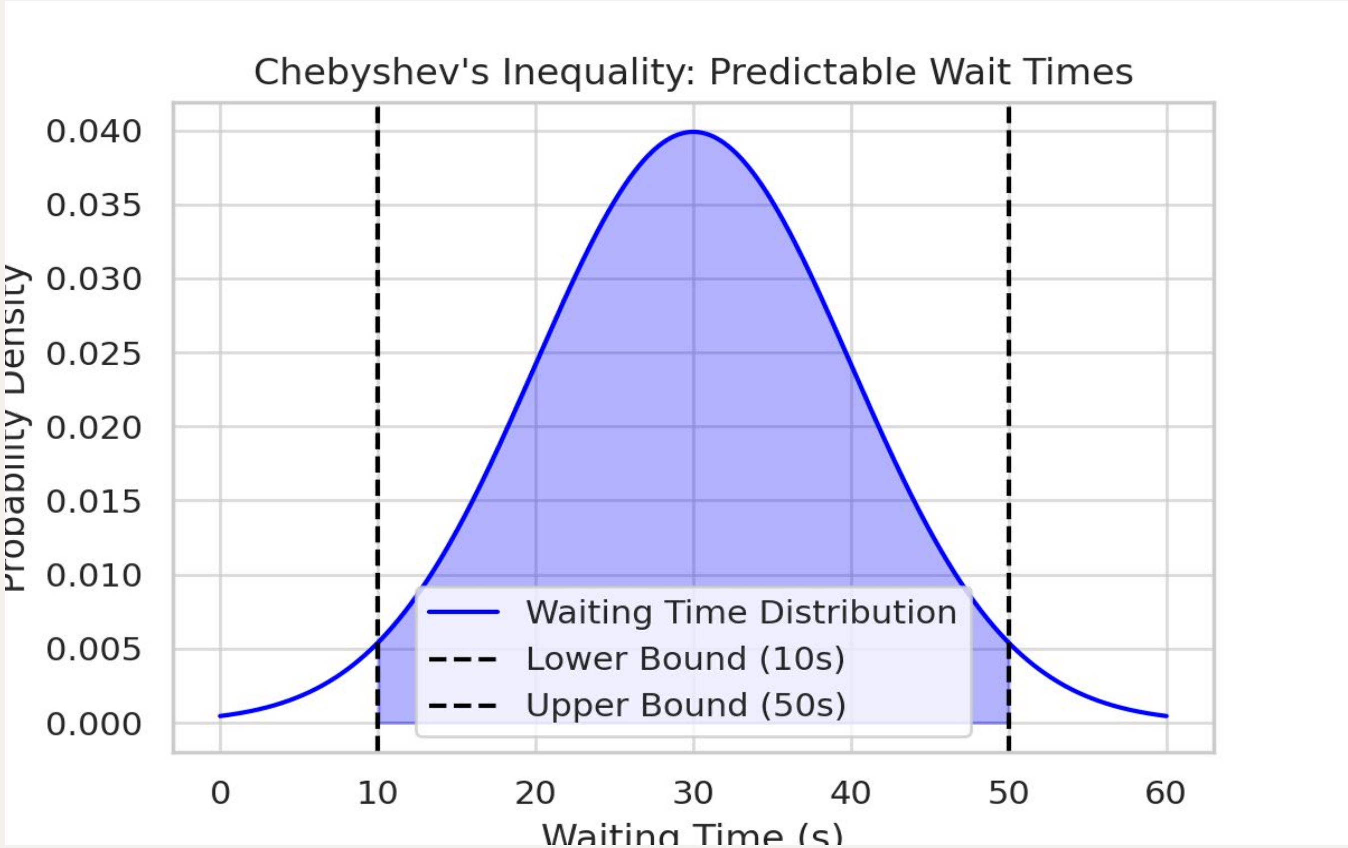
Why Genetic Algorithm and not deterministic approach?

- Handles Large Search Space Efficiently.
- Works Well with Noisy, Unpredictable Data.
- Multi-objective Optimization.
- Gets better over generation.
- Best for offline optimization.

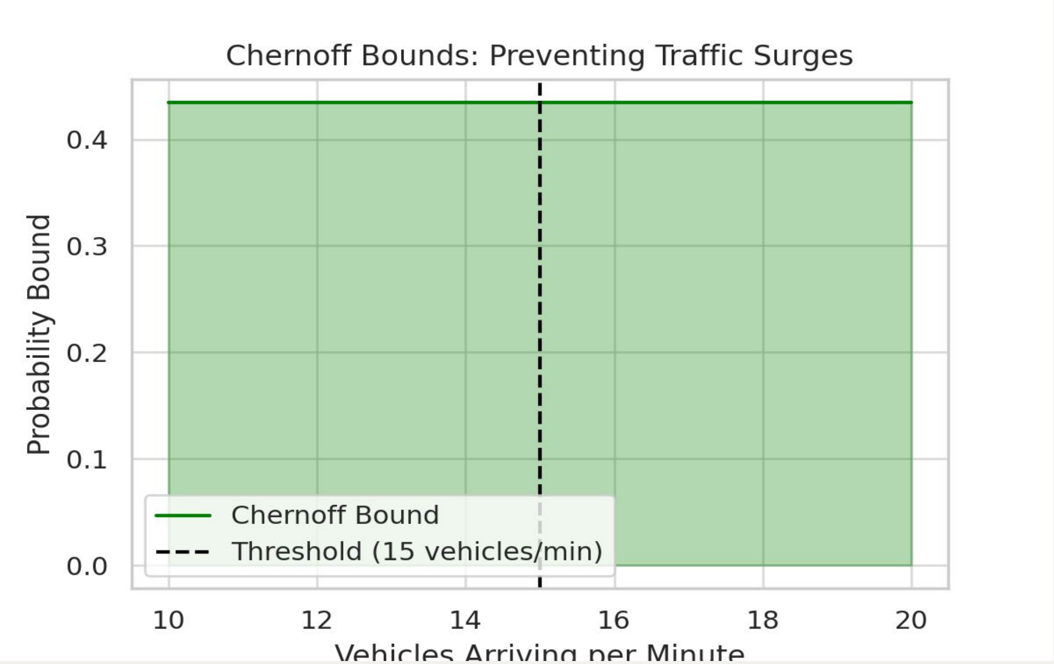



HYBRIDIZE A MODEL OF GENETIC ALGORITHM AND Q-LEARNING.
FUTURE PLANS
INSTALL AN EMERGECY SYSTEM WHICH WORKS ONLY WHEN AMBULANCE IS COMING.
INTEGRATION WITH LIVE TRAFFIC DATA AND MAKE IT ADAPTIVE MODEL.

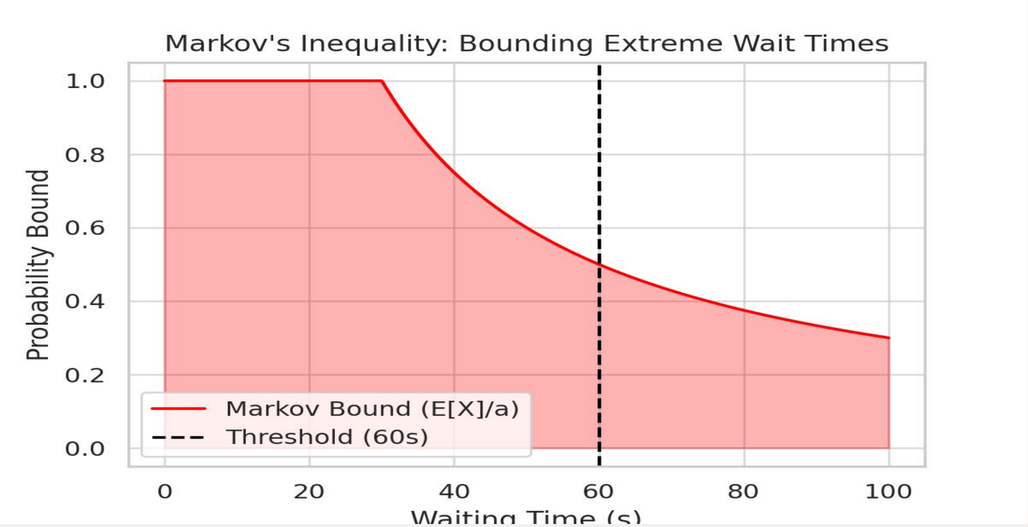
T5: Derivation of Bounds



Chebyshev’s Inequality



Cheroff bounds



Markov’s Inequality

Ahmedabad University

Hevit - AU2340194

1. Markov's Inequality.

Let X be the waiting time in a simulation run. Then:

$$P(X \geq a) \leq \frac{E[X]}{a}$$

- *Use to bound the probability that wait time exceeds a threshold.*

2. Chebyshev's Inequality

- *Use to show that how often will the waiting time be much worse than average.*

3. Chernoff Bounds

- It provides an exponential bound on the probability of extreme events, like unexpected surges in traffic.

References

- Yossidoctor. (2022). AI-Traffic-Lights-Controller/Traffic Control With AI - Project Report.pdf at main · yossidoctor/AI-Traffic-Lights-Controller. GitHub. https://github.com/yossidoctor/AI-Traffic-Lights-Controller/blob/main/Traffic%20Control%20With%20AI%20-%20Project%20Report.pdf

- Dimri, S. C., Indu, R., Bajaj, M., Rathore, R. S., Blazek, V., Dutta, A. K., & Alsubai, S. (2024). Modeling of traffic at a road crossing and optimization of waiting time of the vehicles. Alexandria Engineering Journal, 98,114-129. https://files.campuswire.com/e5a84109-702a-42ef-929f-f6d7c7febdee/e36df6c5-4cd7-48ad-9cb0-0e56caf85d12/1-s2.0-S1110016824004344-main.pdf

- Dharma9696. (2022). *GitHub - dharma9696/Traffic-Lights-Genetic-Algorithm: Code to apply genetic algorithm on traffic lights in SUMO*. GitHub. https://github.com/dharma9696/Traffic-Lights-Genetic-Algorithm

Thank You