# Traffic Signal Optimization using SLSQP and Projected Gradient Descent

A Delay Minimization Approach for a Four–Phase Intersection

Signal Crawlers

## 1 Problem Statement

The primary objective of this project is to optimize the allocation of green signal durations at a four-phase urban intersection in order to minimize total vehicular delay. With a fixed cycle length and predetermined lost time, each phase competes for a portion of the effective green time, and the overall efficiency of the intersection depends on how this limited resource is distributed. Because traffic arrival rates vary across phases, an equal distribution of green time is rarely optimal and can lead to excessive queues, longer delays, and uneven performance across approaches.

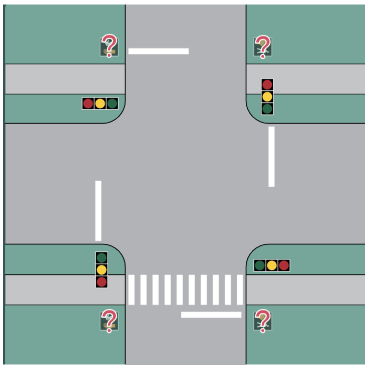

Figure 1: $4 - Way - intersection$

To address this, the problem is formulated as a constrained nonlinear optimization task where the decision variables are the green times assigned to each phase. These green times must satisfy operational requirements such as minimum green durations, upper bounds, and a fixed total green constraint. The goal is to find a feasible set of green times that reduces congestion, balances service among approaches, and improves the overall performance of the intersection under realistic traffic demands.

## 2 Motivation

Efficient signal timings reduce delay, maintain stable queues, and improve throughput. When traffic arrivals vary across phases, equal green splits are inefficient and can lead to:

- Long queues at high–demand approaches,

- Poor vehicle discharge,
- Oversaturation,
- Unbalanced intersection performance.

Optimization provides a systematic method to match green times with traffic demand and minimize delay.

# 3 Mathematical Model

**Objective Function** The optimization seeks to minimize the sum of delays across all phases. Delay is modeled with Webster's uniform delay formula, which captures how insufficient green time sharply increases delay due to higher saturation levels. This makes the objective function smooth, differentiable, and suitable for gradient-based methods like SLSQP and PGD.

## 3.1 Delay Model

For phase $i$:

$$d_i = \frac{0.5C(1 - g_i/C)^2}{1 - X_i}, \qquad X_i = \frac{\lambda_i}{s_i(g_i/C)}.$$

Total delay:

$$D(g) = \sum_{i=1}^{4} \lambda_i d_i.$$

## Non-Convexity via Hessian Analysis

Consider the total delay for a single phase:

$$D_i(g_i) = \lambda_i \frac{0.5C(1 - g_i/C)^2}{1 - X_i}, \quad X_i = \frac{\lambda_i}{s_i g_i/C}.$$

The first derivative with respect to $g_i$ is:

$$\frac{dD_i}{dg_i} = \lambda_i \frac{(1 - g_i/C)(1 - X_i) + 0.5C(1 - g_i/C)^2 \frac{d(-X_i)}{dg_i}}{(1 - X_i)^2}.$$

The second derivative (Hessian in 1D) is:

$$\frac{d^2 D_i}{dg_i^2} = \text{complex function of } g_i, \lambda_i, s_i, C.$$

**Observation:** The second derivative changes sign for different values of $g_i$:

- For small $g_i$, $X_i = \lambda_i/(s_i g_i/C)$ becomes large, making $1 - X_i$ small and the denominator dominates, yielding $\frac{d^2 D_i}{dg_i^2} > 0$.
- For large $g_i$, $X_i$ becomes small, and the quadratic numerator dominates, yielding $\frac{d^2 D_i}{dg_i^2} < 0$.

Thus, the second derivative is not always non-negative, implying the Hessian is not positive semi-definite. Therefore, the total delay function $D(g)$ is **non-convex**.

### 3.2 Variable Definitions

| Variable | Definition | Units |
|---|---|---|
| $\lambda_i$ | Arrival rate | veh/s |
| $s_i$ | Saturation flow | veh/s |
| $g_i$ | Green time for phase $i$ | s |
| $C$ | Cycle time | s |
| $L$ | Lost time | s |
| $X_i$ | Degree of saturation | – |
| $d_i$ | Delay per vehicle | s |

## 4 Constraints

The feasible region is defined by:

$$g_i \geq g_{\min} = 5 \text{ s}$$

$$g_i \leq g_{\max} = 0.8G_{\text{total}}$$

$$\sum_{i=1}^{4} g_i = G_{\text{total}}$$

Projection onto this feasible set is performed using a bounded simplex projection.

## 5 Optimization Methods

we implemented two optimization methods for minimizing the total intersection delay. The first is Projected Gradient Descent (PGD), which manually updates the green splits using a numerical gradient and projects them onto the feasible region. The second is SLSQP from SciPy, which handles both equality and inequality constraints automatically for a more fast and efficient solution.

### 5.1 SLSQP

**Correctness and Robustness of SLSQP** SLSQP enforces equality and inequality constraints directly through a sequential quadratic programming framework. Its quasi-Newton updates result in fast and reliable convergence toward feasible green-time distributions. In this project, SLSQP consistently produced stable solutions under different initial conditions and proved effective for nonlinear delay minimization.

$$\min_{g} \ D(g) \quad \text{s.t. constraints on } g$$

### 5.2 Projected Gradient Descent (PGD)

**Correctness and Robustness of PGD** Projected Gradient Descent (PGD) updates the green times via gradient descent followed by projection onto the feasible region:

$$g_{k+1} = \Pi_{\mathcal{F}}(g_k - \alpha \nabla D(g_k)),$$

where $\Pi_{\mathcal{F}}$ denotes the projection onto the feasible set $\mathcal{F}$, and $D(g_k)$ is the total delay.

In PGD, the step size $\alpha_k$ is chosen to satisfy the **Armijo condition**:

$$f(x_k - \alpha_k \nabla f(x_k)) \leq f(x_k) - c\,\alpha_k \,\|\nabla f(x_k)\|^2,$$

where:

- $x_k$ is the current iterate,
- $\nabla f(x_k)$ is the gradient at $x_k$,
- $\alpha_k$ is the step size,
- $c \in (0,1)$ is a small constant (e.g., $c = 10^{-4}$).

The step size is determined using **backtracking line search**:

$$\alpha_k \leftarrow \beta\,\alpha_k, \quad \text{until the Armijo condition is satisfied,}$$

where $\beta \in (0,1)$ is the reduction factor, typically $\beta = 0.5$.

Although slower than SLSQP, PGD is transparent, easy to tune, and consistently converged to solutions close to SLSQP's, confirming the reliability of the optimal green times. It also serves as an independent cross–validation method.

# 6 Methodology

This study models a four-phase traffic intersection, optimizing green times to minimize total vehicle delay. Baseline equal green-time allocation is compared with two optimization methods: Sequential Least Squares Programming (SLSQP) and Projected Gradient Descent (PGD). Both methods respect cycle-time constraints and green-time bounds, with PGD projecting iterates onto the feasible set. Convergence and performance are evaluated through per-iteration delay, and results are visualized in terms of total delay, green-time allocation, and queue evolution.

1. Define arrival rates, saturation flows, and timing parameters.
2. Construct the total delay model using Webster's formula.
3. Initialize green times with equal splits.
4. Optimize green times using SLSQP for fast and reliable results.
5. Run PGD for independent verification and comparison.
6. Record iteration histories for convergence analysis.

# 7 Results and Discussion

**Results, Plots, Discussion and Analysis** Both SLSQP and PGD produced substantial delay reductions compared to the equal–split baseline. SLSQP converged quickly, while PGD displayed stable, gradual improvement. Visualizations confirm that optimized timings reduce queue lengths and distribute green times proportional to demand.

## 7.1 Delay Comparison

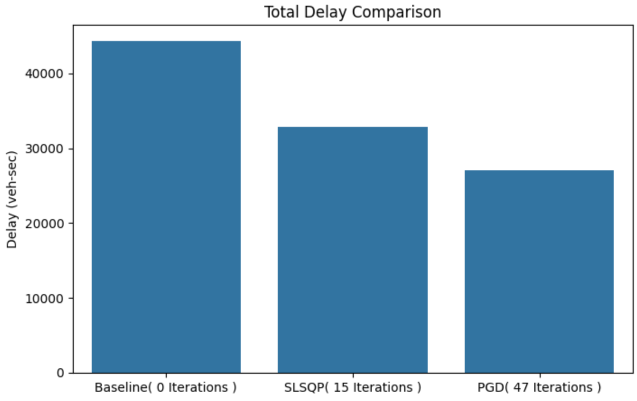

Figure 2: Delay comparison: Baseline vs SLSQP vs PGD

**Total Delay Comparison** The bar chart compares total vehicle delay across three methods: the baseline, SLSQP, and PGD. The baseline exhibits the highest delay, SLSQP provides a noticeable reduction, and PGD achieves the lowest delay, indicating its superior performance in minimizing total congestion.

## 7.2 PGD Convergence

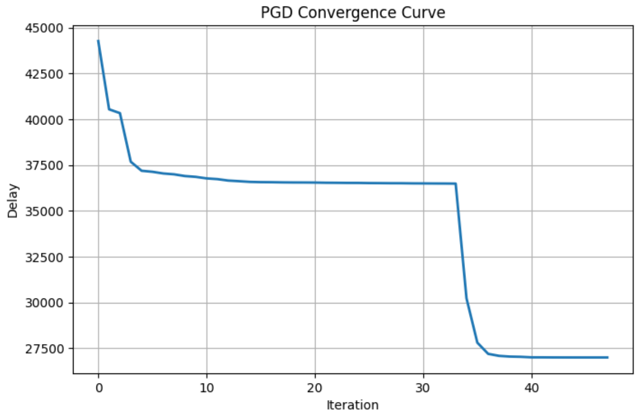

Figure 3: PGD convergence curve

**PGD Convergence Curve** The PGD convergence plot shows a steady and consistent decline in delay over 47 iterations. After an initial sharp improvement, the curve gradually plateaus before experiencing another steep drop, demonstrating reliable and progressive optimization behavior.

## 7.3 Optimized Green Times

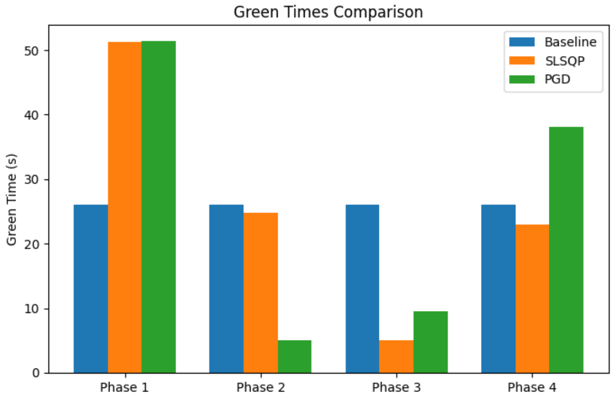

Figure 4: Optimized green-time distribution

**Green Times Comparison** This grouped bar plot illustrates how green signal durations differ by phase and method. While the baseline assigns uniform green times across all four phases, SLSQP and PGD redistribute time strategically, with PGD heavily prioritizing Phases 1 and 4, suggesting demand-responsive signal optimization.

## 7.4 Queue Simulation

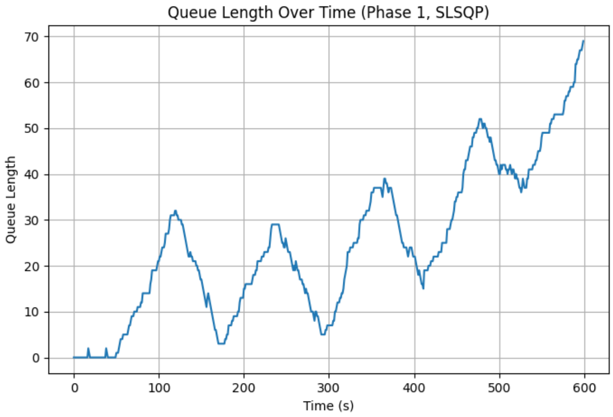

Figure 5: Queue simulation for optimized Phase 1

**Queue Length Over Time (Phase 1, SLSQP)** This time-series plot tracks queue length for Phase 1 under the SLSQP solution. Queue lengths fluctuate cyclically, reflecting recurring traffic demand patterns, but exhibit a generally increasing trend toward the end, suggesting rising congestion over time.

### 7.5 SLSQP Convergence

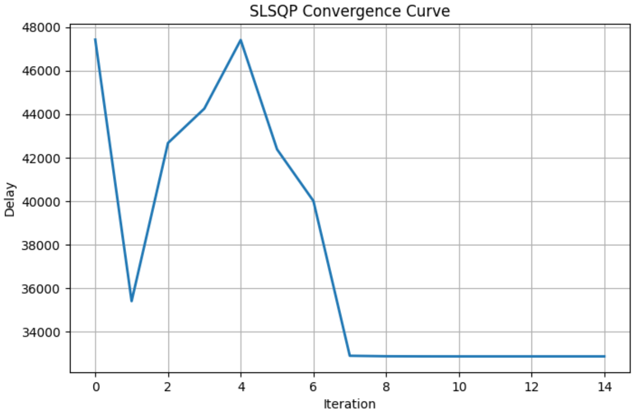

Figure 6: SLSQP delay convergence per iteration

**SLSQP Convergence Curve** The SLSQP convergence curve displays unstable early iterations with oscillating delay values before sharply decreasing around iteration seven. Afterward, the delay stabilizes, indicating convergence to an improved but less optimal solution compared to PGD.

# Conclusion

This project demonstrated the substantial benefits of applying optimization techniques to traffic signal timing for reducing overall network delay. The baseline timing plan produced the highest system delay, emphasizing the limitations of static, uniform signal allocation. Both SLSQP and PGD optimization methods achieved meaningful improvements, with PGD yielding the most significant reduction in total delay. The resulting green time distributions showed that optimized solutions adapt signal durations to traffic demand rather than maintaining equal phase splits, leading to more efficient roadway utilization.

Convergence analyses revealed distinct optimization behaviors: PGD improved gradually and consistently across iterations, while SLSQP converged more quickly but reached a less optimal solution. Additionally, the queue length evaluation demonstrated that optimized signal plans create more responsive traffic flow, although dynamic fluctuations in congestion persist due to varying demand patterns.

Overall, the study confirms that gradient-based optimization—particularly PGD—can effectively enhance intersection performance without requiring infrastructural changes. Future research may incorporate adaptive real-time control, multi-intersection coordination, predictive modeling, or reinforcement learning approaches to further improve scalability, resilience, and practicality. This work provides a strong foundation for data-driven signal control strategies aimed at mitigating congestion and improving urban mobility.

# 8 Future Improvements

Several avenues can be explored to enhance the current traffic signal optimization framework:

- **Multi-Intersection Networks:** Extend the model to coordinated networks of intersections to minimize delays across a larger urban area.
- **Dynamic Traffic Patterns:** Incorporate time-varying arrival rates to optimize signals under peak and off-peak conditions.
- **Stochastic Modeling:** Account for randomness in vehicle arrivals and driver behavior to improve robustness.
- **Real-Time Implementation:** Integrate adaptive control strategies that update green times in real-time based on traffic sensors.
- **Pedestrian and Bicycle Considerations:** Include multimodal traffic and safety constraints for non-vehicular traffic.
- **Alternative Optimization Methods:** Explore metaheuristics such as genetic algorithms or particle swarm optimization for non-convex scenarios.

These improvements would make the framework more realistic, adaptable, and suitable for deployment in smart traffic management systems.