# Traffic signaling using Genetic Algorithms

*Arita Berisha*[1], *Puhizë Doçi*[2]

1 University of Prishtina, Faculty of Electrical and Computer Engineering, Prishtina, Kosova.
arita.berisha@student-uni.pr.edu

2 University of Prishtina, Faculty of Electrical and Computer Engineering, Prishtina, Kosova.
puhize.doci@student.uni-pr.edu

**Keywords**: Traffic Signaling Problem, Genetic Algorithm

## 1. Introduction and Problem Description

Traffic signal scheduling constitutes a fundamental issue in urban planning and transportation engineering [1]. With roots dating back to 1868, traffic control systems have evolved dramatically, yet the primary goal remains the same: ensuring the safe and efficient passage of vehicles through city intersections [2]. In this study, we focus on a practical instance of the Traffic Signaling Problem (TSP), as outlined in the 2021 Google Hash Code competition [3].

This variant involves the allocation of appropriate signal timings at each intersection in a city, bearing in mind a plethora of constraints and objectives [4]. The complexity of this problem arises from the multitude of conflicting interests and constraints [5].

Every traffic light has at least two states, symbolizing 'stop' and 'go' with the universal red and green colors respectively. The challenge lies in devising an optimal traffic light schedule, intended to minimize overall transit time for all vehicles and maximize the number of vehicles reaching their destinations before a pre-specified deadline [1].

The complexity of this problem arises from the multitude of conflicting interests and constraints. Firstly, traffic lights need to synchronize with the planned paths of all cars in the city, requiring an intricate and dynamic balancing act. Moreover, the automatic nature of contemporary traffic systems implies the necessity for careful design and timing, without the possibility for real-time human intervention.

## 2. Solution Method

Inspired by the versatile nature of genetic algorithms in tackling combinatorial optimization problems [6], we devised a Genetic Algorithm (GA) to handle our version of the Traffic Signaling Problem (TSP) [7]. The choice for GA was driven by its robustness and flexibility in navigating large and complex search spaces, such as the one exhibited by the TSP [8].

Our solution space is constructed by a set of schedules that assign green light durations to each street intersection, with the potential for these schedules to violate hard constraints, such as traffic flow requirements or intersection capacities. To tackle this, we have incorporated these violations into our fitness function, assigning them a significantly higher weight compared to soft constraints.

An initial population of solutions is generated randomly, where each solution represents a potential schedule for the traffic lights at every intersection in the city. This population is then evolved over multiple generations, with each generation subject to selection, crossover, and mutation operations, mimicking natural evolutionary processes [10].

1. **Selection**: In this operation, parent solutions are chosen to create offspring for the next generation. We implement tournament selection, where a random subset of the population is selected and the best individual from this subset is chosen as a parent [10].

2. **Crossover**: This operation involves swapping parts of two parent solutions to generate two new offspring solutions [10].

3. **Mutation**: To introduce diversity into the population, the mutation operation randomly alters parts of a solution. Here, the mutation operation may change the green light duration for a randomly selected street at an intersection [10].

4. **Inversion**: This operation, invoked with a certain probability, randomly selects a subset of the schedule, and reverses the order [10].

The quality or fitness of each solution in the population is evaluated based on the total time spent in traffic, with the objective of minimizing this time. The most fit individuals are then selected to form the next generation [9]. This process is repeated until a termination condition is met, such as reaching a maximum number of generations or achieving a solution with acceptable fitness. The best solution found over all generations is then returned as the optimal schedule for traffic lights.

Table 1 shows the details for the genetic algorithm implemented in our approach:

| **Operation** | **Description** |
|---|---|
| Select_with_replacement | Select a random solution from the population for breeding. |
| Crossover | Exchange segments of two parent solutions to produce two offspring. |
| Mutate | Randomly change the green light duration for a randomly selected street. |
| Inversion | Randomly select a subset of the schedule and reverse the order. |

*Table 1. Format of the Genetic Algorithm implemented.*

This solution methodology is implemented in the **genetic_algorithm** function detailed in the provided code. This function encapsulates the key elements of a typical GA, including initialization, evaluation, selection, crossover, and mutation operations. Moreover, it also incorporates some advanced techniques, such as inversion and delta evaluation, which are known to enhance the performance of GAs in complex optimization tasks like the TSP [11].

## 3. Preliminary experimental results

Preliminary results are shown in Table 2, contrasting the best solutions obtained against some of the finest outcomes reported by previously published algorithms. The computational effort for each instance is configured in line with the standards set by the 2021 Google Hash Code competition.

Despite our algorithm still being under active development and experimentation, current results demonstrate significant promise, achieving comparability with state-of-the-art approaches, although falling short of the best-known solutions.

The row titled "Best Available" collates the best solutions known to date, resource taken from Kaggle[1], including instances [1], [2], [3], [4], [5] and [6] in the columns.
The row marked "Us" is reserved for our method's outcomes.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| Best Algorithm | 2002 | 4568819 | 1306213 | 2483375 | 731726 | 1460486 |
| Algorithm 1 | 2000 | 4347500 | 1076198 | - | 434403 | 293170 |
| Algorithm 2 | 2000 | 4565642 | 1231878 | 969685 | 661797 | 455737 |
| Algorithm 3 | 2002 | 4566479 | 1299132 | 1572805 | 707937 | 1291147 |
| Algorithm 4 | 1002 | 4547901 | 1231030 | 344586 | 684514 | 1328301 |
| Us | 1002 | 1502454 | 42052 | - | 321639 | 9141 |

*Table 2. Comparing results of different algorithms with our own*

## 4. Future Work

The research we're starting focuses on enhancing the efficiency of traffic signal scheduling. Initially, we plan on creating new and better methods specifically suited for the Traffic Signaling Problem (TSP) [12]. These new methods will help us come up with a wider variety of better solutions, which will make the genetic algorithm we're using explore more options effectively.

In the future, our research agenda includes looking into different kinds of genetic algorithms like distributed genetic algorithms and co-evolutionary genetic algorithms [13]. Our goal is to use the unique capabilities of these variations, such as their ability to maintain a diversity within the solutions and solve smaller problems that depend on each other. On top of that, we're considering the exploration of other metaheuristic and hybrid methodologies, which might work better than just using a genetic algorithm by itself on this problem.

## References:


1. Tan, M.K. *et al.* (2016) 'Optimization of urban traffic network signalization using genetic algorithm', *2016 IEEE Conference on Open Systems (ICOS)* [Preprint]. doi:10.1109/icos.2016.7881994.
2. Henry, J.J., Farges, J.L. and Tuffal, J. (1983) 'The prodyn real time traffic algorithm', *IFAC Proceedings Volumes*, 16(4), pp. 305–310. doi:10.1016/s1474-6670(17)62577-1.
3. Google Hash Code, "Competition Problem Statement", Google Hash Code, 2021.
4. De Schutter, B. and De Moor, B. (1998) 'Optimal Traffic Light Control for a single intersection', *European Journal of Control*, 4(3), pp. 260–276. doi:10.1016/s0947-3580(98)70119-0.
5. Srinivasan, D., Choy, M.C. and Cheu, R.L. (2006) 'Neural networks for real-time Traffic Signal Control', *IEEE Transactions on Intelligent Transportation Systems*, 7(3), pp. 261–272. doi:10.1109/tits.2006.874716.
6. 'Genetic algorithms in search, optimization, and machine learning' (1989) *Choice Reviews Online*, 27(02). doi:10.5860/choice.27-0936.
7. de Luca, S. *et al.* (2017) 'Transportation systems with connected and non-connected vehicles: Optimal Traffic Control', *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)* [Preprint]. doi:10.1109/mtits.2017.8005660.
8. Xin Yao, Yong Liu and Guangming Lin (1999) 'Evolutionary programming made faster', *IEEE Transactions on Evolutionary Computation*, 3(2), pp. 82–102. doi:10.1109/4235.771163.


[1] https://www.kaggle.com/competitions/hashcode-2021-oqr-extension/overview

9. De Schutter, B. (2000) ‘Optimal control of a class of linear hybrid systems with saturation’, *SIAM Journal on Control and Optimization*, 39(3), pp. 835–851. doi:10.1137/s0363012999354648.
10. Holland, J.H. (1992) *Adaptation in natural and artificial systems* [Preprint]. doi:10.7551/mitpress/1090.001.0001.
11. Sastry, K., Goldberg, D.E. and Kendall, G. (2013) ‘Genetic algorithms’, *Search Methodologies*, pp. 93–117. doi:10.1007/978-1-4614-6940-7_4.
12. Jin, Y., Sendhoff, B. and Körner, E. (no date a) ‘Simultaneous generation of accurate and interpretable neural network classifiers’, *Multi-Objective Machine Learning*, pp. 291–312. doi:10.1007/3-540-33019-4_13.
13. Cantú-Paz, E. (2001) ‘Efficient and accurate parallel genetic algorithms’, *Genetic Algorithms and Evolutionary Computation* [Preprint]. doi:10.1007/978-1-4615-4369-5.