Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2024)
4-6 November 2024, Male, Maldives

2024 4th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) | 979-8-3503-9118-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICECCME62383.2024.10796302

A Tabu Search and Genetic Algorithm for traffic light scheduling*

Naouel Talbi
LAOTI Laboratory
University of Jijel.
Jijel, Algeria
naouel.talbi@hotmail.com

Ismahane Souici
LAOTI Laboratory
University of Jijel.
Jijel, Algeria
souici.ismahane@yahoo.fr

Ahmed Alioua
LaRIA Laboratory
University of Jijel.
Jijel, Algeria
ahmed.alioua@univ-jijel.dz

***Abstract*—The actual traffic light system with its prefixed timers suffer from critical traffic jams due to several factors, including the significant amount of vehicles increasing in the cities. This situation requires a more complex signal control, and hence, a more dynamic scheduling. In this paper, we propose two separated metaheuristic: the Tabu Search Algorithm (TSA) and Genetic Algorithm (GA) for the optimization of the traffic signal in order to compare them and seek for the fittest traffic lights strategy (phases and timings) suitable for a road network with multiple intersections. Employing a urban simulator, both approaches are analysed and their results compared in terms of the following key metrics: Number of vehicles passing through the intersections, the global waiting time and trip time.**

***Index Terms*—tabu search, TSA, genetic algorithm, GA, optimization, traffic light**

I. Introduction

The population density in the cities has recently increased more quickly than the past few decades. It has been recorded that approximately 57.5% of the world's population now reside in urban areas. By 2050, world urbanization is expected to increase to 72% [1].

This process of urbanization has significantly improved people's standards of living by establishing water supplies, medical facilities, residential and commercial structures, and accessible transportation. However, this high rate of rural migration and intensive use of natural resources is severely affecting the environment, and causing various problems such as waste management, energy, and increasing massively the amount of vehicles which results in traffic congestion.

The actual traffic light control system, with its preset timing is more likely to increase the vehicles waiting time and causes more traffic congestion. Therefore, one of the most long-standing challenges in the traffic and transportation fields is how to optimize the traffic light control strategy to increase traffic efficiency and decrease vehicle waiting time and trip time.

The tradition techniques used in the traffic light control mainly include a fixed-time control. In other terms, it has predefined light phases and duration based on traffic flow estimation. Although, this strategy perform relatively well in confined and regular traffic flows thus it responds poorly for real-time traffic conditions.

Studies has shown that using a smart traffic management system can significantly reduce the traffic congestion by the use of, for instance, Wireless Sensor Networks (WSN), Radio-Frequency Identification (RFID), Global Positioning System (GPS) and Internet of Things (IoT) for amassing real-time traffic data [2].

As a first contribution, we propose two meta-heuristics: the Genetic Algorithm (GA) and Tabu Search Algorithm (TSA) that are known to be fit for complex optimization problems and avoid local optima with large search space to optimise the traffic light control system for a road network of multiple intersections with several directions. This optimization refers to better signal phases and timings that leads to increase the number of passing vehicles and decrease the waiting time. With these outcomes, we compare both algorithms to see which one is fit for our problem.

The remaining of this paper is organized as follows. In section 2, we present a brief literature review. In section 3, we summarize the general idea for the proposed model. Sections 4 and 5 describe, respectively, our proposed GA and TSA based traffic light control system and section 6 discusses the simulation obtained results. And finally, the conclusion is presented in section 7.

II. LITERATURE REVIEW

Many paradigms and technologies were applied in research these past years to solve the traffic congestion problem. Applying machine learning algorithms such as reinforcement learning where the authors in [3] proposed a method to solve the static light times allocations, which are disregardful of the current traffic flow. The current phase of the traffic intersection, the length of the queue and the maximum waiting time for a vehicle in each lane was used as inputs for the neural network.

Likewise, the authors in [4] proposed a distributed deep reinforcement learning without a central server which consists on local learning and global consensus.

Other researchers used metaheuristics like: GA [5], where the goal was to increase the number of vehicles passing through the traffic signal system in one intersection with four lanes. The researchers used a simple representation for their solution using 4 bits (genes), one bit for each lane (1 for green light and 0 for red light). This method gave a significant enhancement in number of vehicles leaving the intersection compared to previous works, however it is applied on a simple intersection with a fixed cycle time.

Besides genetic algorithm, other metaheuristics as the particle swarm optimization (PSO) [6] and ant colony algorithms [7] gave good results to resolve the traffic light management system due to its capability to seek for the the best solution in a large search space. Which lead to the application of hybrid algorithm based on metaheuristics and other methods such in [8], where they applied the genetic algorithm and fuzzy logic on a more complex road network with four intersections. The solution was a chromosome with 16 genes (bits), each one of the 16 bits corresponds to a light signal.

Some other papers proposed also a hybrid algorithm as in [9] where they combined GA with Machine Learning (ML) resulting into one single framework (BGA-ML) for more efficiency.

To summarize, a large amount of works in the field of smart traffic light system has been published these past years given good results compared to the ancient models. However, most of these systems are for simple road networks that involves one intersection and/or for limited roads.

III. METHODOLOGY

A. Road Network design

The road network model considered for this study is a constitution of four intersections labeled as Inter.1, Inter.2, Inter.3 and Inter.4 as shown in Fig.1. Each intersection has four way, having three lanes with three possible direction: turn-right, straight-forward, and turn-left.

B. Assumptions

- We do not consider driver behaviour, including factors such as vehicle movement delays and overly fast habits.
- We do not consider emergency cases.
- No yellow light is adhered, only green and red lights. The yellow light here is not considered as significant for the road safety since the traffic lights device display the green and red light duration.

C. Conflict tables

For our research, we need to gather additional information about the road network such as the non-conflict movements in the intersections for each road segment. Each possible configuration will be represented by a 4×3 binary table with four paths (N,S,E,W) and three possible directions: right-turn, straight and left-turn. The Fig.2 illustrate an example of a possible configuration of the green light in an intersection where '1' refers to a green light and '0' for a red one. For instance, the lane in the incoming road for north and south

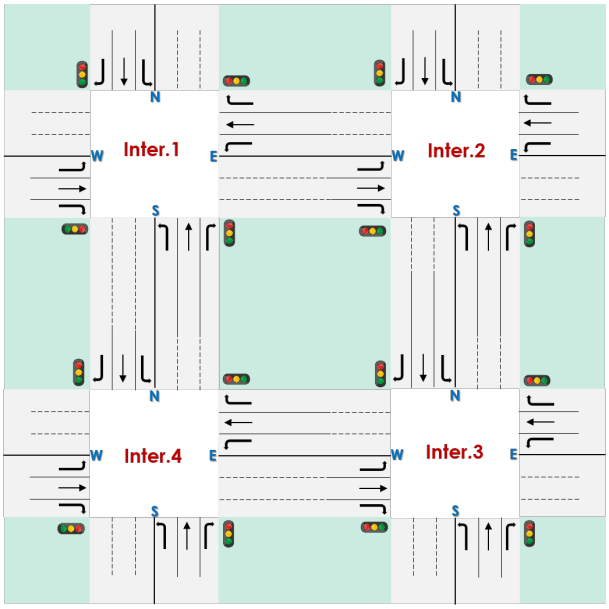

Fig. 1. Intersection architecture

Road/Lane	Right	Straight	Left
N	0	0	1
E	1	0	0
S	0	0	1
W	1	0	0

Fig. 2. Example of a conflict table

of the intersection have the green light for turning left so the vehicles are free to pass as the same time as the east and west roads for the right turning.

IV. GA FOR TRAFFIC LIGHT SCHEDULING

A. The Genetic algorithm

The Genetic Algorithm is a meta-heuristic based on Charle Darwin's theory of natural evolution. It hold a population of individuals (chromosomes) which represents potential solutions to the problem and repeatedly update these individuals for a number of generations, until a particular stopping criteria is satisfied. Based on the hypothesis of natural selection, at each generation, the fittest individual is selected for reproduction in order to produce the next generation, the offspring resulted will be subject to mutation and crossover procedures. Thus, ensuring the exchange between the characteristics of the parents to better solutions, as outlined in Algorithm 1.

Algorithm 1: GA pseudo-code

1. population= Initialize_Population();
2. fitness= Evaluate(population);
3. **For** generation=1 : MaxGenerations **do**

4. [parents, parents_fitness]= Selection(population, fitness);
5. offspring= Crossover(parents, parents_fitness);
6. offspring= Mutation(offspring);
7. new_fitness= Evaluate(offspring);
8. elites= Elitism(population, fitness, offspring, new_fitness);
9. [population, fitness]= Update_population(elites, offspring, fitness, new_fitness);
10. best_solutions= Update_best(population, fitness);
11. **End**

The algorithm describes the GA pseudo-code. It begins with the first population initialization using Initialize_Population function which generates a number of potential solutions as a starting point which includes both the timing and the state of a phase. These parameters are randomly initialized under certain criteria such as the range for the phase timing. After these individuals are computed, each one of them is evaluated (Evaluate(Population)). Next, a group of individuals among the population is selected (Selection(Population, fitness)) based on their calculated fitness value to be parents for the next population. The first reproduction operator used is the crossover (Crossover(parents, parents_fitness)), where the new individuals are formed by swapping a sub-sequence from each parent with another using crossover points. These recently produced offspring are now subjected to the mutation operator by flipping one gene with another. This ensure a more diverse population. Once the reproduction process is over, the mutated offspring are evaluated (Evaluate(offspring)). Elitism procedure is called to extract a handful of good chromosomes from the previous population so that the best individuals are not lost in the reproduction cycle. Now that the new population is successfully formed (Update_population()), a best solution for each generation is chosen/updated according to its fitness value and this process (from line 4 to 10) is repeated during a maximum number of generations.

The subsections below describes our approach for the solution encoding and the fitness function.

B. Genetic representation of the solution

GA manages a population of potential solutions called chromosomes. Coding a chromosome is the first step and it is accomplished by means of a vector of integers representing the phase state and its timing. The Fig.3 depict a chromosome with an example. There is four genes for each intersection, and each gene is divided into a constant number of phases: each phase contain a state and its timing. The state is obtained from one of the conflict tables previously mentioned. 8 is the number of phases we chose to set our model on. For instance, the first phase of the intersection "1" has a state with the ID **1** which corresponds to the conflict table in Fig.2 and **50** seconds as the duration of the phase. For further clarification, the first phase of the traffic light system will set a 50 seconds green

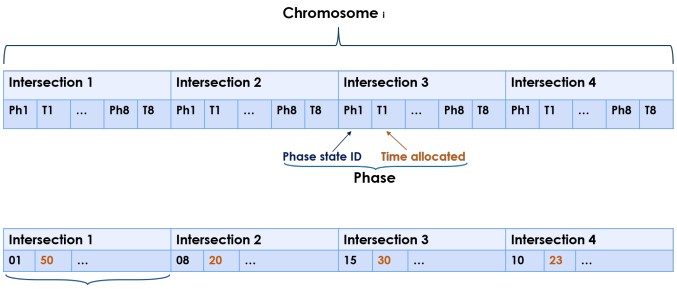

Fig. 3. A chromosome representation

light for vehicles in the north and south road to turn only left and for the east and west roads to turn only right.

C. Fitness function

Once the population of chromosomes is initialized, each solution is evaluated as mentioned previously in the GA pseudo-code using the following fitness function in Eq.1, which metrics are extracted from the simulation's events.

$$fitness(chromosome_i) = \frac{Nvp_i^2}{(SD \times Wt_i) + Trip_i} \quad (1)$$

- Nvp: is the number of vehicles that pass and reach their destination during the simulation time.
- Wt: is the global waiting time of all vehicles during the simulation.
- $Trip$: is the global trip time of all vehicles during the simulation.
- SD: the simulation duration.

In Eq.1, we focus on increasing the number of passing vehicles (Nvp^2), the value is squared in order to prioritize it over other metrics. On the other hand, the waiting time (Wt) and the global trip time must be minimized. Wt is multiplied by the simulation duration in order to prioritize its reducing over the global trip time ($Trip$).

V. TSA FOR TRAFFIC LIGHT SCHEDULING

A. The tabu search algorithm

Tabu search is a meta-heuristic method originally proposed by Fred Glover in 1986. It is an optimization technology based on imposing restrictions that guides a local heuristic search procedure to explore the solution space beyond local optimality [10]. These restrictions are designed as forbidden moves, or more likely, 'tabu'. The word tabu stands for a strategy related to restrict or forbid certain choices. The Algorithm 2 is a pseudo-code for the tabu search methodology. First, the list containing the restrictions known as tabu list is initialized empty, and a current solution is initialized as well, the solutions here have the same representation as the chromosome mentioned before in the GA algorithm. Next, for a maximum number of iterations, a new group is generated from the current solution by repeatedly switching around the solution's characteristics until a variety of possible solutions emerge, called neighbors. Then each neighbor; as long as it

is not listed as tabu, is evaluated and based on its results, the neighbor with the best evaluation is selected and defined as the current solution to be used for the next iteration. In the meantime, the tabu list is updated. Finally, at the end of each iteration, the best solution is updated according to their calculated evaluations in order to find the optimal solution. The objective function is similar to the GA fitness function.

Algorithm 2: TSA pseudo-code

```
1.   tabuList=[];
2.   current_solution= initialSolution();
3.   For iteration=1 : MaxIterations do
4.       neighbors= generateNeighbors(current_solution);
5.       If (neighbor ∉ tabuList) then
6.           evaluations= evaluateNeighbor(neighbors);
11.      End
7.       next_solution= select_solution(neighbors,
         evaluations);
8.       tabuList= update_tabuList(tabuList, neighbors);
7.       current_solution= next_solution;
9.       update_BestSolution()
11.  End
```

B. Tabu list

The tabu list contributes in the search's diversification by preventing the algorithm from cycling back to recently visited solution and avoid local optima. In our algorithm, the tabu list strategy is quite simple. At every iteration, the current solution and its neighbors are listed as 'tabu' to get away from solutions in the search space already explored in the previous iterations. In addition, it allow a more diversification for the neighbors in order to enhance the local search.

VI. SIMULATION

In this section, we will evaluate the performance of our algorithms based on the multiple intersections schema mentioned before, using MATLAB [11] for the GA and TSA programming and SUMO (Simulation of Urban MObility) framework [12]) for our model simulation. First, we explain briefly the preparation for the files needed for the simulation. Then we present the results and compare the two approaches based on their objective function values and metrics as the number of passing vehicles, the global waiting time and the global trip time.

A. Environment and methodology

Since testing a traffic light control model is expensive and challenging in real life, a multitude of simulation tools such as SUMO have been introduced the past few years. This framework has been frequently employed to evaluate the performance of a traffic light logic. In this work, four files are needed to establish the sumo simulation, two files are predefined for the road network and the vehicle's route.

After the initial solutions in GA and TSA are generated in MATLAB, the phase state and the duration are extracted from the solutions and then placed in a third file for the traffic light plan called additional file. These files will serve as inputs for the configuration file. For each chromosome in every generation, a new configuration file along with the additional file generated.

To recap, the required files are listed below:

- **Input files**
  - Network file(network.net.xml): contains all the road network characteristics.
  - Route file(route.rou.xml): contains the vehicles routes and travel direction.
  - Additional file(additional.add.xml): contains the traffic light logic.
- **Configuration file**
  - Configuration file(configuration.sumocfg): for the SUMO simulation run.
- **Output files**
  - Tripinfo file(trip.xml): contains the vehicle's trip information.

Once the configuration file is launched, the simulation starts and a trip information file is generated by SUMO. From this output file, we will extract the metrics mentioned previously in the paper (number of passing vehicles, waiting time...) to calculate the GA and TSA evaluation. In every generation, the best solution will be updated for both algorithms until a stopping criteria is reached.

TABLE I.
SIMULATION PARAMETERS USED IN OUR EXPERIMENTS

GA	**TSA**	**value**
Phase state	Phase state	Rand[1,16].
Phase duration	Phase duration	Rand[10,30]
Population size	/	40
/	Neighbors size	32
Generations	Iterations	25
Crossover Rate	/	0.8
Mutation Rate	/	0.3
Elitism Rate	/	0.2
Simulation duration	Simulation duration	500s
/	Maximum tabu size	300

B. Parameters

In our program, the stopping criteria is a fixed number of iterations/generations as shown in Table I alongside with other required parameters.

- The phase state is represented by an ID, chosen randomly between [1-16], each number corresponds to one of the 16 predefined conflict tables.
- The phase duration is the time allocated to a phase and it is randomly chosen between two values.
- The size of the chromosome population and the neighbors is 32.
- Number of iterations for both GA and TSA are 30.
- The tabu list can hold 300 as maximum number of prohibited solutions.

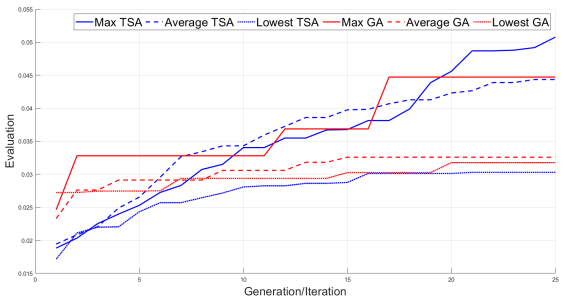

Fig. 4. Convergence of TSA and GA for 20 runs for each algorithm

TABLE II.
TSA AND GA SIMULATION RESULTS (OBJECTIVE FUNCTION VALUES) FOR 500S

Evaluation	Algorithm	
	TSA	**GA**
Best evaluation	0.0508	0.0447
Average evaluation	0.0432	0.0355
Worst evaluation	0.0303	0.0318

C. Results and discussion

In our scenario, we have stored the data gathered from each simulation run for both GA and TSA. The evaluations of the best solutions for each iteration are presented in Table II. It compares the genetic algorithm with the tabu search algorithm for the proposed model in terms of maximum, average and lowest computed objective values of 20 executions for both algorithms after an execution of 25 iterations for each algorithm.

The simulations are performed by SUMO for 500s with continuous entrance for the vehicles into the intersections. It is evident that TSA obtains better outcomes in terms of objective values, even if the results for GA gives what we consider as satisfactory results. However, considering Fig.4 where the best, the average and the lowest evaluations are extracted from both TSA and GA results, TSA gave better results for the three measurements. .

The main purpose is to make the road traffic smoother, and thus, increasing the number of vehicles that leaves the intersections and decreasing the global waiting time and trip time. So based on the average evaluation for both algorithms, we extracted the previous mentioned metrics from the trip information file and presented the results in Table III and Fig.5. The figure clearly shows that TSA allow more vehicles to finish their trips. Obviously, the waiting time is better as well since it is 20.9% less than in TSA, as well as for the travel time with a difference of 10.2%. These results implies that TSA is capable of finding a better configuration for the traffic light logic (phase configuration and duration) by enhancing the traffic flow.

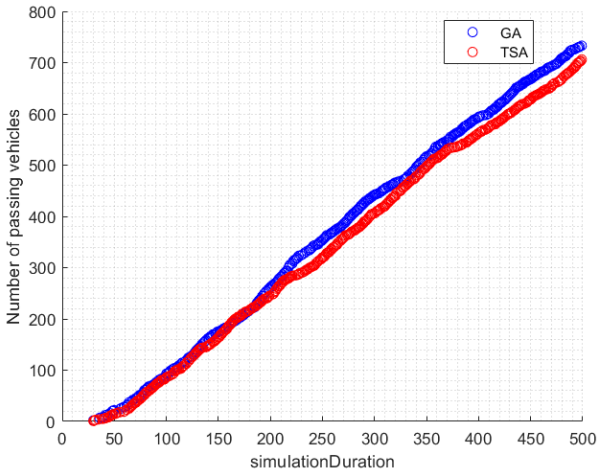

Fig. 5. Number of vehicles finishing their trip for TSA versus GA for 500s

TABLE III.
TSA AND GA SIMULATION RESULTS (EVALUATION METRICS: NUMBER OF PASSING VEHICLES, GLOBAL WAITING TIME AND TRIP TIME) FOR 500S

Metrics	Algorithm	
	TSA	**GA**
Number of passing vehicles	733	706
global waiting time	24119	30473
global trip time	56562	62994

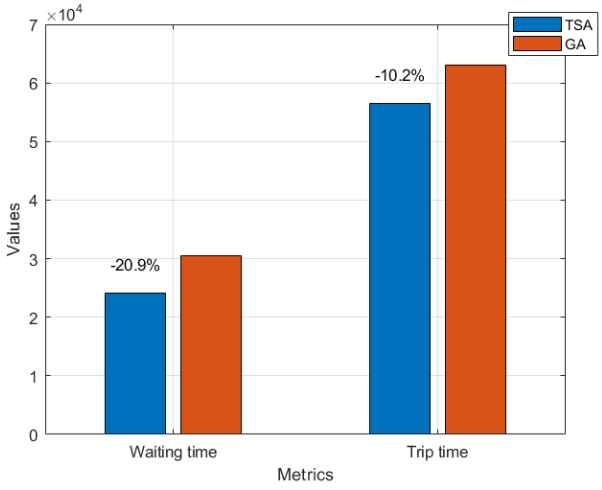

Fig. 6. Global waiting time and global trip time for TSA and GA (simulation of 500s)

VII. Conclusion

This paper proposed two meta-heuristics approaches in order to compare them, a Genetic Algorithm and Tabu Search Algorithm for the traffic light control scheduling problem. The main objective was to increase the number of passing vehicles and decrease the global waiting time and trip time by searching for the best light signal configuration. After determining the solution representation and the objective function suitable for the evaluation for both algorithms, we used MATLAB for the programming and SUMO for the simulation launch. From the previous results, the following deductions are extracted:

- The proposed TSA give successful outcomes and a significant improvement in the traffic flow compared to GA.
- It takes less iteration for TSA to converge to GA best results.
- The diminutive global wait time and trip time for the TSA is more significant than in GA which makes the TSA traffic light strategy more effective.
- The number of passing vehicles in TSA is more important than in GA.
- The tabu list in the searching algorithm prevent from local optima and hence, explore a larger search area for better solutions.

This study demonstrates that the TSA provides good strategies for the traffic light scheduling problem. However, There are certain challenges in the application of these two meta-heuristcs such as the computation time. To find a satisfactory solution may take a lot of iterations and evaluations through a large search space. Furthermore, each evaluation must be preceded by a simulation run. In this case, by SUMO which take an important computation time. For now, it is only limited for a traffic simulation scenario which means a preset number of vehicles for a fixed duration, not with the real live traffic. As future work, we will seek for a more sophisticated model by decreasing the computation time and using data from a real traffic scenario. And hopefully, increase even more the number of vehicles passing through the road network and decreasing the waiting time and trip time.

References

[1] K. Davis, "The urbanization of the human population," in *The city reader*, pp. 43–53, Routledge, 2015.

[2] H. M. Sherif, M. A. Shedid, and S. A. Senbel, "Real time traffic accident detection system using wireless sensor network," in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 59–64, IEEE, 2014.

[3] M. B. Natafgi, M. Osman, A. S. Haidar, and L. Hamandi, "Smart traffic light system using machine learning," in *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pp. 1–6, IEEE, 2018.

[4] B. Liu and Z. Ding, "A distributed deep reinforcement learning method for traffic light control," *Neurocomputing*, vol. 490, pp. 390–399, 2022.

[5] A. Tamimi, M. Abunaser, A. A. Tawalbeh, and K. Saleh, "Intelligent traffic light based on genetic algorithm," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 851–854, IEEE, 2019.

[6] J. García-Nieto, E. Alba, and A. C. Olivera, "Swarm intelligence for traffic light scheduling: Application to real urban areas," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, pp. 274–283, 2012.

[7] J. He and Z. Hou, "Ant colony algorithm for traffic signal timing optimization," *Advances in Engineering Software*, vol. 43, no. 1, pp. 14–18, 2012.

[8] S. M. Odeh, "Hybrid algorithm: fuzzy logic-genetic algorithm on traffic light intelligent system," in *2015 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pp. 1–7, IEEE, 2015.

[9] T. Mao, A.-S. Mihăită, F. Chen, and H. L. Vu, "Boosted genetic algorithm using machine learning for traffic control optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7112–7141, 2021.

[10] F. Glover, V. Campos, and R. Martí, "Tabu search tutorial. a graph drawing application," *Top*, vol. 29, no. 2, pp. 319–350, 2021.

[11] T. M. Inc., "Matlab version: 23.2.0 (r2023b)," 2023.

[12] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.