Optimization of urban semaphore times turning into JSSP

Antonio M R Almeida[1], Jose A F Macedo[1], and Javam C Machado[1]

Federal University of Ceara, Fortaleza CE 60440-900, Brazil
manoel.ribeiro@lsbd.ufc.br, jose.macedo@dc.ufc.br,
javam.machado@dc.ufc.br http://www.lsbd.ufc.br

Abstract. The objective of this paper is to cover the research in the area of adaptive traffic control with emphasis on applied optimization methods. A distinction can be made between classical systems, which operate with a common cycle time, and the more flexible ones, phase-based approaches, which are shown to be more suitable for adaptive traffic control. Classic optimization solutions for this problem result in a model which is relatively easy to represent but may be difficult to fit into the standard mixed-integer programming (MIP) scheme. We propose an alternative approach to find an optimal global solution for the green wave problem on hot routes, which consists of reducing it to a Job Shop Scheduler problem using the Webster Model to adapt the cycles to road characteristics and average traffic speed.

Keywords: Trajectory Pattern Mining, Green Wave, Traffic-Light Scheduler, Job Shop Scheduler

1 INTRODUCTION

In recent years, there has been a significant increase in the number of systems with built-in GPS function. The proliferation of these devices allowed the collection of a large number of GPS trajectories. Many applications, such as route planners, route finders, and georeferenced social networks, began using GPS data information to provide new services that were not previously available.

Traffic congestion is one of the leading causes of productivity loss and decrease of living standards in urban settings. The data generated by smartphones with GPS provide rich datasets that can be mined to offer solutions to urban traffic problems. Our problem creates an optimization method to program traffic lights that encourages the creation of convoys of vehicles forming a green wave of routes with greater flow, using as input the flow variation estimated by GPS trajectories as input.

Traffic congestion causes delays that add significant costs to society and business on a daily basis and also increases CO^2 emissions and the risk of accidents. The study [9] reports that in 2002, people who spent 100,000 hours in queues on Greater Copenhagen's road infrastructure, generated an economic loss of more than 750 million euros.

In order to alleviate traffic congestion, there are basically two solutions: improving public transportation or expanding road infrastructure. In urban areas, the latter is often impossible due to residential areas adjacent to existing roads. A more subtle way to improve network performance is to make better use of existing roads, which can be achieved in part by proper configuration, optimization of traffic signal parameters, increasing the overall flow efficiency of vehicles and reducing losses.[19]

This paper is an extended version and clarify the novel scientific content presented in the manuscript GPS2GR [3] of the same authors.

2 BACKGROUND

2.1 Green Wave Problem

Mobility is one of the most significant concerns in the world's largest cities. Due to the growth vehicle fleet levels and the limitation of urban roads, It can be assumed that the vast majority of vehicles exceed the flowing capacity of large cities. Paper [6] studied the optimization of urban traffic using an Automaton cellular traffic model. This study demonstrates the effect of signal control strategies on vehicle traffic. In addition, it has shown that traffic controlled by traffic lights can be reduced to a simpler single-line problem.

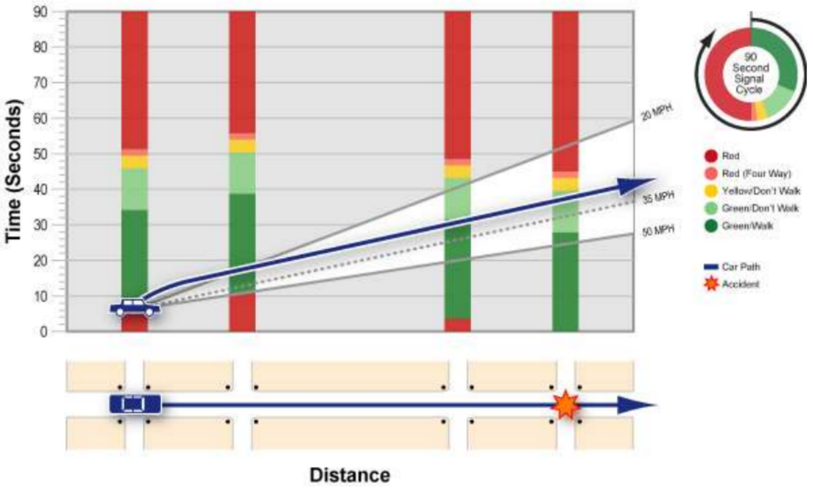

Fig. 1. Example of a Green Wave in the road

Another study [18] investigated traffic flow controlled by traffic lights on a single track route using the ideal speed model. previous studies differ in the relationship between road capacity and the saturation limit. The [16] study described vehicle traffic through a sequence of traffic lights on a highway and presented a dynamic model in which a vehicle is prevented from passing other vehicles.

Figure 1 has an illustration with an example of a graph with a vehicle in the space and time dimensions and the constraints for the green wave to occur. In this the vehicle must be maintained between a minimum and maximum average speed, thus maintaining the synchrony of displacement in the physical space of the track and the elapsed time

The definition of the problem consider the flow of vehicles passing through a finite series of traffic lights. Each vehicle moves at an average speed v, the path is not blocked by other vehicles and there are no non-synchronized traffic signals. We consider vehicle traffic in the one-dimensional network. It encourages the formation of vehicle trains through the orchestration of traffic lights where vehicles tend to travel together, all with the average speed of the v route. In synchronized strategy, all traffic lights change simultaneously from red to green, and vice versa over a fixed period of time. The sum of the times is called the cycle time.

In the green-wave strategy, the traffic light changes with a certain time delay t shifted between the semaphore phases of two successive intersections. The delay time is called the travel time. The change of traffic lights spreads backwards, like a green wave.

When a vehicle arrives at a traffic light and the light is red, the vehicle stops there. Then, when the traffic light changes from red to green, the vehicle moves on. On the other hand, when a vehicle reaches a traffic light and the light is green, it does not stop and keep moving without changing the speed (green wave).

2.2 Systems for Offline Optimization

Definition 1 ***(Hot Route)*** *It is used to reveal the common behavior of a set of paths in a given time window. The hot route in this context is the sequence of adjacent edges that share the certain amount of traffic (minTraffic) defined as a parameter. [13]*

Definition 2 ***(Convoy)*** *Grouping of vehicles in space-time following a common flow at the average track speed held together in a green-wave window.*

Definition 3 ***(Webster model)*** *Study on semaphore times and formulas that defines a simplified model of relation between the traffic lights times. [20]*

Definition 4 ***(Green Route)*** *Is a Green Wave over a Hot Route. Green Route has the advantage of optimizing the flow independent of the path that the flow follows in that time window.*

A lot of research has been done since the 1960s on offline traffic optimization systems. In order for the mathematical model to be solvable without an external evaluation tool, such as TRANSYT, it must contain functional constraints that relate, for example. the appropriate displacement for traffic lights along arterial parameters for the specific flow, such as the saturation flow rate and the vehicle convoy dispersion factor. Considering a typical optimization formulation involving a common cycle time, divisions and green compensations, even considering a discretization of time, the search space of a solution is very broad, especially when considering some form of simulation to obtain the target value. Systems that operate in this way are best suited for off-line use, because trial runs have resulted in very costly processes, typically of several hours, even for small networks.

2.3 Adaptive & Cooperative Systems

The adaptive traffic control systems aim to coordinate the controlled intersections by signal, controlling a performance index that indicates an optimized solution, but possibly not optimal. They do this by dynamically adjusting cycle times, phase sequences, and green time according to both detected and predicted traffic, thus reacting to dynamic aspects of traffic that cannot be captured by the static optimization routines used to generate time-of-day plans. Here we cite three adaptive operating systems: RHODES [15], Phase-by-Phase [14] and DOGS [12]. These proposals offer solutions that converge at a time t to a minimum that can be local. Simplifications commonly used as standard cycle times may not be suitable for large networks.

2.4 Traffic density-based discovery of hot routes in road networks

This work [13] proposes a new density-based algorithm named FlowScan. Instead of clustering the moving objects, road segments are clustered based on the density of common traffic they share.

2.5 The Job-Shop Scheduling Problem (JSSP)

The problem Job-Shop Scheduling is classified in the literature as NP-Hard, that is, it is a problem for which there are no algorithms that solve it at polynomial time. It is a combinatorial optimization problem (Souza, 2001), which is an explicit or implicit enumeration of all possible alternatives to search for satisfactory solutions or the search for the optimal solution, depending on the class of the optimization algorithm used.

The optimization algorithms can be classified into two classes: mathematical models (search for optimal solution) and heuristics (search for approximation).

Since the problem of optimizing green waves on public roads is of NP-Hard complexity, the solutions have been adaptive methods [19] to find local solutions through trial and fee-feedback, similar to the methods of signal processing. In figure 2 it has a runtime Pi that corresponds in the original problem to the green time of the traffic light and a time to start Ti that corresponds to the time that the convoy will take to reach the traffic light.

Consider a set of n tasks (jobs) and a set of m machines, where each task is composed by a set of operations whose execution order is defined. For each

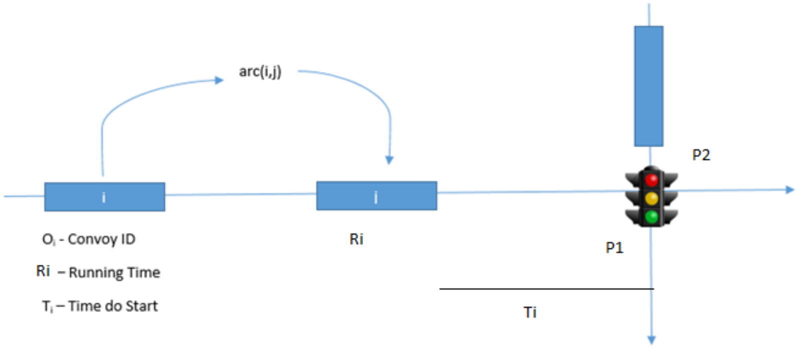

Fig. 2. Job Scheduler problem for green wave

operation, we want to know in which machine it must be executed and the necessary time for it to be completed. Each convoy becomes a task, which runs on multiple machines (traffic lights) synchronously.

- A task cannot be assigned to the same machine twice.
- There is no order of execution in relation to different task operations.
- An operation cannot be interrupted.
- Each machine can process only one task (operation) at a time.

The purpose of JSSP is to distribute tasks among machines so that the processing time necessary to finish executing all of them (also known as makespan) is as small as possible. In other words, we want to determine the sequence of machines that minimize the makespan.

In our approach, we made a reduction of the problem of the green wave to a problem of Job Scheduler where the traffic lights are the processors and the convoys of vehicles are the tasks.

2.6 Simulation of Urban MObility

Simulation of Urban MObility (SUMO)[4] is an urban traffic microsimulation framework, including import modeling and demand generation components. SUMO helps to investigate various research topics such as route selection algorithms and traffic signal programming or vehicle communication simulation. Therefore, the framework is used in different projects to simulate automatic driving or traffic management strategies.

In our work, we use the SUMO traffic simulation mechanism to compare the base-line with constant traffic times, but optimized against the traffic-light times produced by the GR algorithm.

3 METHODOLOGY

Our approach provides a flow optimization solution on previously identified Hot Routes. We propose a method to optimize convoys of vehicles on hot routes. These clusters of vehicles will gain benefit from the green light exchange, when they are approaching an intersection. However, these convoys of vehicles need to have an average speed that matches the average speed of the hot route. We have concluded our approach by offering an alternative way to solve the optimization problem, reducing it to a job shop scheduler problem, and applying a branch & bound method to solve it.

In short, the proposed GR algorithm produces a green wave over predefined flows using combinatorial optimization approach to arrive at the best solution for an urban context.

3.1 Webster model

Variable	Definition
T_d	Drive reaction time
T_y	Yellow time
T_{gr}	Global Red time
T_r	Red time
T_g	Green time
T_c	Cycle time
T_f	Clearance time

Table 1. Webster model for Traffic Lights

According to the Webster model in the table 1 for urban semaphores [20] , the yellow time of an isolated semaphore is given by the equation 1. For usual situations, we can consider the driver's reaction time $T_d = 1s$ and an average deceleration given by $a = 2.8m/s$. Therefore, the yellow time depends on the velocity in the approaching pathway (V).

$$T_y = T_d + \frac{V}{2a} \tag{1}$$

For simplicity the standard defined in the table 2 is adopted for T_y.

The general red time T_{gr}, see table 3, of an approximation is a function of the following variables:

- Velocity V of this approximation;
- Width L of the track to be crossed (of the distance to be covered by the Vehicle within approach);
- Vehicle c length;

Approaching pathway	T_y
Street - 40Km/h	3s
Avenue - 60Km/h	4s
Expressway - 80Km/h	5s

Table 2. Yellow time

- Clearance time T_f (minimum time elapsed between the beginning of time of the transverse track and the moment of entry of a vehicle into Speed).

Considering the usual situations, the following parameters were adopted:

- Secondary roads (streets) with a maximum width of 9 m, and speed of 40 km / h
- Preferred routes (avenues) up to 30 m wide, and speed 60 km / h;
- Vehicle of 5 m;
- Clearance time $T_f = 1.2s$.

Approaching pathway	Way to be crossed	T_{gr}
Any	Street	$0s$
Street	Avenue	$2s$
Avenue	Avenue	$1s$
Expressway	Expressway or Avenue	$0.4s \rightarrow 1s$

Table 3. General red time

According to Webster's model, we must adopt sub-multiple cycles of time or programming period, to enable any change of plans without cutting abruptly the time of green current, never adopt T_c less than 30 s and only in special cases accept T_c greater than 120 s

$$T_c = T_g + T_y + T_r + T_{gr}, 30 \leq T_c \leq 120 \tag{2}$$

The concept of minimum green time T_{g-min} is associated with the concept of time-wasted of the Webster model and must be respected under the penalty of causing major traffic jams. In table 4, the usual values of T_{g-min} are shown:

To define the percentage of the cycle that corresponds to green, we adopted a proportion of traffic volume between the approach path and the secondary pathway, that is, where F_a is the approach path flow and F_b the secondary path flow, then

$$T_g = T_c - T_y - T_{gr} * T_a/(T_a + T_b) \tag{3}$$

$$T_c = T_r + T_y + T_{gr} + T_g \tag{4}$$

Approaching pathway	minimum green time T_{g-min}
Street - 40Km/h	12s
Avenue - 60Km/h	15s
Expressway - 80Km/h	17s

Table 4. Minimum green time

3.2 Implementation of Green Route

The classic problem of optimizing [19] the semaphore scheduler to produce a green wave effect on a path is represented by the following mathematical model, see notation in table 5:

$$\begin{aligned} \min \quad TT(\Psi, q^*(\Psi)) = \sum_{l=1}^{L} q_l \cdot t_j(\Psi, q^*(\Psi)) \\ subject \quad to: C_{min,n} \leq C_n \leq C_{max,n}, \\ 0 \leq \theta_n \leq C_n - 1 \quad \forall_n, \\ \phi_{min,np} \leq \phi_{np} \leq \phi_{max,np} \quad \forall_{n,p}, \\ C_n = \sum_{p=1}^{P_n} (\phi_{np} + I_{np}) \quad \forall_n \end{aligned} \tag{5}$$

$n \in \{1, ..., N\}$	Intersection indexes
$p \in \{1, ..., P_n\}$	Indexes of phases at intersection n
Ψ	A set of signal timing settins
C_n	Cycle time for intersection n
θ_n	Offset of intersection n
ϕ_{np}	Phase p green time for intersection n
I_{np}	interstage (lost) time from the end of phase p until the next
$l \in \{1, ..., P_n\}$	Link indexes
$q_l \in \mathbf{q} * (\mathbf{\Psi})$	User equilibrium link flow given signal timing parameters
$t_l(\Psi, \mathbf{q} * (\mathbf{\Psi}))$	Travel time on link l considering signal timings and user response

Table 5. Notation for the traffic signal and network model

We propose an optimized method that reduces the green-wave problem to the Job-shop scheduler problem, allowing the use of Branch and Bound solver [10] to find an optimal solution to the original problem.

We generate convoys on the hot routes and look for an optimal solution for these convoys to travel on hot routes without colliding. Inspired by the work [11] we propose the formation of virtual convoys of vehicles on a hot route, optimizing the displacement of these convoys along the hot route. With this model the sequence of block subsections traversed by a convoy are viewed as a set of machines in a job shop scheduling problem, where convoys correspond to jobs. The traversing of a block subsection by a convoy is called an operation o_i, and requires a running time p_i. [8]

3.3 Reduction of the green wave problem to Job-shop scheduler

Consider the following transformations from the original semaphore optimization problem to a JSSP:

- Traffic lights are mapped to the processors
- Convoys are formed to replicate the Webster cycle on the Hot Route
- Convoys are mapped to the Jobs
- Green time is the job run time
- Webster model are problem constraints
- We will use the branch and bound method to find an optimal solution that minimizes makespan
- The solution generates a set of Jobs that can execute without conflicts in the processors, that is, a set of convoys of vehicles that travel at the average speed of the route and are able to pass in all the open semaphores

In this way the JSSP variables are:

- **Job-shop(J_m)**: In a job shop with m machines (traffic lights), each job represents a car convoy that has a particular and predetermined route to follow (hot route)
- **Processing time (p_{ij})**: The p_{ij} represents the processing time of the Job j on machine i.
- **Makespan (C_{max})**: The makespan (manufacturing time), defined as Max $(C_1, ..., C_n)$, the end time of the last job leaves the system. A minimum manufacturing time usually implies maximum use of the machine (s).

3.4 Implementation details

The whole implementation was done in C# language, because some of the baseline algorithms were originally written in this language. The implemented code is available in the GitHub repository. [2]. The input of GR algorithm are Route Routes and Map (OSM) and the output is the traffic light scheduller (XML).

4 SIMULATION RESULTS

4.1 Input data

For testing purposes, we used real trajectory data extracted from Taxi Simples, a taxi company based in Fortaleza, Brazil. [5] This dataset had a very high

ALGORITHM 1: GR - Green Way over Hot Route

```
Input: $R[]$ set of trajectories with flow annotation
Output: S set of scheduler for all traffic light
$Processors \leftarrow \{\}$;
$Jobs \leftarrow \{\}$;
$w \leftarrow$WebsterModel(R[]);
$w.greentime \leftarrow w.mingreentime$;
(w.greentime++ <= w.maxgreentime) $Processors \leftarrow$ DTL(R[], w);
$Jobs \leftarrow$ MC(R[], w);
$s \leftarrow$JSSP.Solver(Processors, Jobs);
if $s != null$ then
    break;
end
/* repeats by increasing the green time, in search of a solution   */
return $s$;
/* $s$ variable contains an optimal solution                        */
```

ALGORITHM 2: MC-MakeConvoys

```
Input: $R[]$ set of trajectories with flow annotation
Input: w Webster model variables
Output: Jobs list of tasks (convoys)
$Convoys \leftarrow \{\}$;
$ConvoysSize \leftarrow w.getTotalTimeRoute()\ /\ w.CycleTime$;
while $n \neq ConvoysSize$ do
    $c \leftarrow new\ Convoy()$;
    $c.hotRoute \leftarrow this$;
    $c.processingTime \leftarrow w.GreenTime$;
    $c.timeInHotRoute \leftarrow w.CycleTime$;
    $Convoys.add(c)$;
end
return $Convoys$;
```

ALGORITHM 3: DTL-DiscoveryTrafficLights

```
Input: $R[]$ set of trajectories with flow annotation
Input: w Webster model variables
Output: Jobs list of processors (traffic lights)
$Processors \leftarrow \{\}$;
foreach $s \leftarrow R.Segments$ do
    foreach $n \leftarrow s.Nodes$ do
        $Found \leftarrow True$;
        if $n.TrafficSignal == True$ then
            $t \leftarrow new\ Processor(n.Id)$;
            foreach $p \leftarrow Processors$ do
                $Result \leftarrow$checkIfProcessorCloser(p, t);
                if $t.Id \neq p.Id \wedge Result > 30$ then
                    $Found \leftarrow False$;
                end
            end
            if $!Found$ then
                $p \leftarrow new\ Processor()$;
                $Processors \leftarrow p$;
            end
        end
    end
end
```

frequency sampling rate, but was filtered down to reflect one with a low-sampling character. Maps in OSM XML format were obtained from OpenStreetMap and then parsed into a road network-like graph data structure using an XML parser written in C# [22, 1] . The dataset used has collected data paths in the period between 11/14/2015 and 09/05/2016, containing 22,791,003 GPS points.

4.2 Experiments Setup

For the experiments, we used a Intel (R) Core (TM) i7-4770S CPU @ 3.10GHz with 16GB RAM memory, 256KB Cache L1, 1.0 MB Cache L2 and 8.0 MB Cache L3.

The trajectory data of the Taxi [5] were downloaded and unzipped, and the day 6/5/2016 was randomly chosen to perform the experiments of this article, which alone contains 749,049 GPS trajectory points. The format of the files was converted from CSV to GPX, and the city map of Fortaleza was downloaded from the website Open Street Map [17].

To demonstrate the effectiveness of our algorithms, we perform the sequence of the following steps on the raw data of GPS trajectories in the city of Fortaleza.

4.3 Experimental Results

To evaluate the GR algorithm we used frequent routes obtained from the data set of taxi trajectories mentioned above, using routes that contained traffic lights in their route and where there were intersections. To extract the Hot routes we use the Flowscan algorithm[13], we use the minTraffic = 1 parameters to ensure that the Hot Routes have the greatest possible coverage of the trajectories of the city. We used the Webster Model to calculate cycle times based on highway type parameters and average track speed.

All the approaches used in this study showed very promising results and high applicability. Based on real-time trajectories of raw data we can obtain the city traffic behavior at each time slice where the behavior is linear. The query visualization can be used as a tool to discover the higher density flow (hot routes) and influence the creation of public policies for urban traffic.

In our experiments, we used the Simulation of Urban Mobility (SUMO) [7] software (version 0.29.0). That tool is open source, and has a tremendous efficiency on large sets of data routes. Moreover, to show the effectiveness of the Green Route algorithm, three simulations were performed on a Intel(R) Core(TM) i7-4770S CPU @ 3.10GHz.

In each simulation, we isolated a fixed number of hot routes, using one in the first experiment, two in the second, and finally three on the third. In addition, for each hot route, we extracted its respective traffic lights, latitude and longitude positions. Furthermore, in order to execute the experiments, we had to calculate the rate of cars passing on each hot routes. For that, each hot route has an attribute called traffic segment, which tells us the number of vehicles in that specific segment. Therefore, by summing up all the traffic segments (on each hot route), we get the rate of cars.

On SUMO, for each simulation, we measured the average trip length (s), the average trip duration (s), and the average waiting time (s). For each simulation, we first performed it using the default traffic lights of the SUMO software, which are generated based on the common traffic of the route. Only after, we performed it again but then using our traffic lights configuration, in which we get it from our GR algorithm.

In Table 6, we present the results of the simulations for the first, second, and third scenarios. In the first line, we show the amount of vehicles loaded, then in the second line the number of hot routes, and finally on the third line the is show lane length (km). The remaining ones are the measures that we used throughout the simulation.

Listing 1.1. Part of scheduler file from GR

```
<?xml version="1.0" encoding="UTF-8"?>
<scheduler>
  <traffic-light id="0">
    <green start="00:00:00" end="00:00:17" />
    <green start="00:00:34" end="00:00:51" />
    <green start="00:01:08" end="00:01:25" />
```

Simulation	First scenario			Second scenario		
	SUMO	GPS2GR	%	SUMO	GPS2GR	%
loaded vehicles #	791	791		6291	6291	
rot houtes #	1	1		3	3	
total lane length (km)	84,94	84,94		653,26	653,26	
Average time						
Trip[1] waiting time (s)	11,6	11,32	-2,41	28,49	28,11	-1,33
Trip duration (s)	110,95	110,53	-0,38	247,5	246,74	-0,31
Trip time loss (s)	34,2	33,8	-1,17	77,71	77	-0,91

Table 6. Simulation results

```
    <green start="00:01:42" end="00:01:59" />
    ...
    <green start="23:59:03" end="23:59:00" />
  </traffic-light>
</scheduler>
```

5 DISCUSSION AND CONCLUSIONS

We were able to complete an implementation and tests of the GR algorithms. This algorithm converged and generated optimized solutions of the opening and closing times of the traffic lights along the hot routes [21].

We prove by simulation that the solution obtained by the GR algorithm is optimal since its optimal solution is always better than the good solutions generated by heuristic methods. Optimization on rot route can solve classic vehicle containment problems at the end of a green wave, as it takes into account flow optimization, that is, green wave overflow, no matter what its path is.

As future work we can consider the application of GR algorithm on real time information streaming in order to reflect real time aspects of the flow such as accidents, road maintenance, etc.

References

1. Almeida, A., Lima, M., Macedo, J., Machado, J.: Dmm: A distributed map-matching algorithm using the mapreduce paradigm (2016)
2. Almeida, A.M.R.: `https://github.com/antoniomralmeida/TPM` (2016)
3. Almeida, A.M.R., Leite, J.L.A., Macedo, J.A.F., Machado, J.C.: Gps2gr: Optimized urban green routes based on gps trajectories. In: Proceedings of the 8th ACM SIGSPATIAL Workshop on GeoStreaming. pp. 39–48. IWGS'17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3148160.3148167, `http://doi.acm.org/10.1145/3148160.3148167`
4. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: Sumo–simulation of urban mobility: an overview. In: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind (2011)

5. Braga, M.: Taxi dataset fortaleza, ceará. `https://s3.amazonaws.com/txsimples_tracking/index.html` (2016)
6. Brockfeld, E., Barlovic, R., Schadschneider, A., Schreckenberg, M.: Optimizing traffic lights in a cellular automaton model for city traffic. Physical Review E **64**(5), 056132 (2001)
7. Center, G.A.: Simulation of urban mobility. `http://sumo.dlr.de/index.html` (2017)
8. Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M.: Evaluation of green wave policy in real-time railway traffic management. Transportation Research Part C: Emerging Technologies **17**(6), 607–616 (2009)
9. Daganzo, C.F., Sheffi, Y.: On stochastic models of traffic assignment. Transportation science **11**(3), 253–274 (1977)
10. Google: Google optimization tools. `https://developers.google.com/optimization/` (2017)
11. Higgins, A., Kozan, E., Ferreira, L.: Optimal scheduling of trains on a single line track. Transportation research part B: Methodological **30**(2), 147–161 (1996)
12. Lauritzen, S.M.: Evaluation of dogs. Danish Road Directorate (2004)
13. Li, X., Han, J., Lee, J.G., Gonzalez, H.: Traffic density-based discovery of hot routes in road networks. In: International Symposium on Spatial and Temporal Databases. pp. 441–459. Springer (2007)
14. Machemehl, R., Shenoda, M.: Development of a phase-by-phase, arrival-based, delay-optimized adaptive traffic signal control methodology with metaheuristic search. Tech. rep. (2007)
15. Mirchandani, P., Wang, F.Y.: Rhodes to intelligent transportation systems. IEEE Intelligent Systems **20**(1), 10–15 (2005)
16. Nagatani, T.: Vehicular traffic through a sequence of green-wave lights. Physica A: Statistical Mechanics and its Applications **380**, 503–511 (2007)
17. OpenStreetMap: Openstreetmap. `http://www.openstreetmap.com/` (2016)
18. Sasaki, M., Nagatani, T.: Transition and saturation of traffic flow controlled by traffic lights. Physica A: Statistical Mechanics and its Applications **325**(3), 531–546 (2003)
19. Warberg, A., Larsen, J., Jørgensen, R.M.: Green wave traffic optimization-a survey. Tech. rep., Informatics and Mathematical Modelling (2008)
20. Webster, F., Cobbe, B.: Traffic signals. road research technical paper no. 56. HMSQ, London **111** (1966)
21. Ying, J.J.C., Shi, B.N., Lan, K.C., Tseng, V.S.: Spatial-temporal mining for urban map-matching (2014)
22. Zheng, Y., Lou, Y., Zhang, C., Xie, X.: Map-matching for low-sampling-rate gps trajectories (Feb 25 2010), uS Patent App. 12/712,857