An Evolutionary Based Approach for the Traffic Lights Optimization Problem

Ivan Davydov[1,2](✉) and Daniil Tolstykh[2]

[1] Sobolev Institute of Mathematics, Novosibirsk, Russia
vann.davydov@gmail.com
[2] Department of Mechanics and Mathematics, Novosibirsk State University, Novosibirsk, Russia
daniil.tolstykh.1996@gmail.com
https://www.researchgate.net/profile/Ivan_Davydov

Abstract. We consider the traffic lights optimization problem which arises in city management due to continuously growing traffic. Given a road network and predictions (or statistical data) about the traffic flows through the arcs of this network the problem is to define the offsets and phase length for each traffic light in order to improve the overall quality of the service. The latter can be defined through a number of criteria, such as average speed, average trip duration, total waiting time etc. For this problem, we present an evolutionary based heuristic approach. We use a simulation model on the basis of the SUMO modeling system to evaluate the quality of obtained solutions. The results of numerical experiments on real data confirm the efficiency of the proposed approach.

Keywords: Simulation modeling · Evolutionary algorithm · SUMO · Traffics lights sheduling

1 Introduction

Due to the continuous growth of traffic in urban area a number of various problems arise. In order to avoid serious congestions on the roads only a few solutions may be applied. Among them we can mark three most effective. The first one is to restrict the possibility of personal car owners to enter the urban area of the city. It can be done in different ways: reduced parking space, paid parking in the central area, paid entrance to the central area on working days, etc. Although this is a very efficient way to reduce the traffic jams and the pollution level, this is usually quite unpopular step for the citizens. It can only be considered in case of a perfectly organized public transportation system, as it should provide the carrying service almost equal to a private one. The second approach consists in the development of the road network. New crossings, bridges, roads, additional

Supported by RFBR according to the research project 19-01-00562.

I. Bykadorov et al. (Eds.): MOTOR 2019, CCIS 1090, pp. 19–29, 2019.
https://doi.org/10.1007/978-3-030-33394-2_2

lanes can significantly reduce the congestion. On the cons this approach is quite expensive. Also it can hardly be applied in the historical centers of big cities, with its narrow streets and buildings standing close to each other. The last but not the least way to improve the quality of the city road network is to perform a fine tuning of all of its existing components. This includes a possible redefining of the routes and lane connections on the intersections, assigning special lanes, prohibiting parking in rush hours etc. One out of the most effective steps in this area is the precise tuning of the traffic lights (TL for short). While this procedure is almost free of charge, if made in an optimal way, it provides quite efficient results. Depending on the initial setup and current load of the network in some cases total travel time might be reduced by 20% and even more.

A number of studies have been done on this topic. Majority of the papers can be roughly divided into three sets. The first ones consider the problem of optimal schedule for isolated crossing. In [13] authors propose a dual step approach for fine tuning of TL on the intersection. Using a number of mobility patterns on intersection an off-line scheme is applied first. The resulting optimal schedules are used then in the on-line settings. In [4] it is supposed that flow is unstable and may be different from hour to hour. To overcome this problem, the authors intend to estimate the quality of a traffic light schedule according to the worst case traffic scenario. It is assumed that for each lane the maximal and the minimal flow values are known, and the total deviation from the median values is bounded from above. A proposed dynamic programming approach allows finding an optimal schedule, although the computational time can exceed 10 hours on an average PC. Solution evaluation is made according to the model, described in Highway Capacity Manual [8]. Synchronization of such isolated crossings is sometimes considered as a problem itself. In [11] authors propose a differential evolution approach and investigate the benefits of parallelism for this complex problem. In [12] authors propose two models to tackle traffic signal coordination problems for long arterials and grid networks.

The second direction of research deals with more complicated systems which consist of several intersection. In [3] a cell-type road network is considered (although the approach can be generalized). It is assumed that the traffic loads are known and fixed. Green lights and offsets for each traffic light are under optimization together with the cycle length, unique for all TL objects. To tackle the problem the authors propose a heuristic approach, based on a Bee Colony algorithm. Although the authors claim the effectiveness of the approach, the tests were implemented only on artificially generated data, so it is impossible to compare the solution provided by the approach with the real life behaviour. The common point in all such works is a formulation of the problem in terms of mathematical programming. It means that the quality of the road network is somehow measured via explicit functions, while the possibility itself to create a model, truthful enough, to simulate the real traffic is quite doubtful.

In order to overcome this difficulty, one can use a simulation model. In general a simulation model is a kind of "black box" for a goal function calculation. Given an input, it performs a number of calculations, and provides clear and understandable output. A concept of simulation modelling is widely applied nowadays in operations research, when the problem under consideration deals with complex systems with many agents. The reason is that inside a sophisticated simulation model a lot of different relationships between the agents of the problem might be incorporated. Being applied to the traffic problems, this approach allows simulating every vehicle, pedestrian, lane of the road, traffic light independently from the other objects, assign unique properties to this object and provide a detailed specification of its interaction with other objects. The result is a microscopic simulation, which can easily be tuned in order to provide simulation as close to reality as needed. Among other simulation tools, SUMO - Simulation of Urban Mobility [10] appears to be both quite accessible and popular among researchers. In [7] the authors propose an evolution approach, based on a Particle Swarm heuristic to tackle the traffic light optimization problem. The criterion to be optimized is the function, which depends on a number of flow measurements, like the number of cars that have reached their destination within the simulation scenario, total travel time, total delay, etc. The components of this criterion were obtained during the SUMO simulation run. Test results on the road networks of Malaga and Sevilla showed that a reasonable tuning of the traffic lights may bring up to 15% raise in the efficiency of the road network.

In this paper we continue the efforts aimed at producing a good union of evolution approaches and SUMO. In order to rise the outcome of the simulation model we have incorporated a set of detectors into the simulated road network. These detectors provide additional data on the traffic flows. This information is then used to make a local improvement of the solution during the search process.

The paper is organized as follows. The second section provides a detailed formulation of the problem. The description of the proposed PSO-based approach is given in Sect. 3. Section 4 contains the results of the computational experiment. Section 5 concludes the research and provides the directions for future investigations.

2 Problem Formulation

We consider the traffic light optimization problem as follows. We are given a road network and an information about the traffic flows. The road network is described via set of arcs and edges. The road network is presented in a OpenStreetMaps format, and contains all the information, including the number of lanes, road marking, crosswalks, etc. There is also a set of traffic light objects given. It is assumed that the number of phases, and their order are predescribed and fixed for every TL. The values of the flows are presented as a statistical data and contain the information on the type, speed and the number of vehicles, that are passing through the measuring sensor. We distinguish between two types ov vehicles: long heavy trucks and private cars. The instance of the problem (scenario) consists of a road network with specified intersection, traffic lights, lane connections, crosswalks and road signs. The second term is a set of vehicles. Each vehicle has its own predefined route and a specified departure time. A traffic lights schedule defines the number of the vehicles that cross the intersections and their order. The aim is to define a schedule for the traffic lights, specifying the length of the green light for each phase of each traffic light, cycle length and phase shift time (offset) in order to minimize the cost function.

We assume that each intersection is controlled by one traffic light, although physically there might be more than one device per crossing. Also, it is assumed that the number of different states of TL and its order is fixed. Thus, the setting of an isolated TL can be encoded as an integer vector. The number of components of this vector corresponds to the number of phases. Each phase itself is also a k-vector, where k is the number of connections, controlled by a particular TL. Each connection represents a possible direction of movement. An example, provided in Fig. 1 demonstrates the concept. Here the number of connections is $k = 12$, and they are ordered in a clockwise direction. During the first phase eight of them are "green" while the other four are "red". Thus each phase of the TL can be denoted by a string of k characters. "GGggrrrGGGg" denotes the phase in which 4 connections have a green light, next 4 are red, and the other 4 are green once again. The difference between the small "g" and the capital "G" denotes that, although both connections are allowed and "green", vehicles moving along "G" connections have higher priority. Phase durations, written in a specified order, like $60, 6, 31, 6, 30, 6$ define the regime of the whole TL object. Here 6 phases follow one after another in a cycle. Thus, the length of a whole cycle is defined as a sum of its phases (139 s for the case). While the cycle length can be excluded from the set of variables, it is not the case for the offset. The latter defines the shift between the beginning of cycles of different TL objects. This option is highly likely to be used, while managing big systems, since it introduces another level of interaction between the TL objects. The set of variables related to one intersection consist of the lengths of the phases and the offset value. Solution of the whole problem can be encoded as a vector which contains an ordered list of all TL's phases and offsets, one for each TL. We assume that both the phase length and the offset can take only integer values. The minimal value of the greenred phase is limited by 8 s due to the safety reasons. The maximal value of the cycle is also bounded, and that induces the boundaries on the length of each phase. The offset value is between zero and the total cycle time. While the choice of the offsets is usually considered as a stand-alone problem, it might cause an inappropriate interrelation between the crossings, so in this model finding the offset values is incorporated into an optimization process.

The second part of the problem which should be defined as well is the optimization criterion. A number of different parameters can be used to estimate the efficiency of the TL schedule. Among them are the number of cars that have reached their destination within a scenario, total travel time and total delay. In [7] the authors propose the following fitness function as a measure of quality:

$$fitness = \frac{TT + SW + (NV \cdot ST)}{V^2 + P}.$$

Here TT denotes total travel time of the vehicles, ST stands for simulation time. SW represents the amount of time that vehicles had to spend waiting on

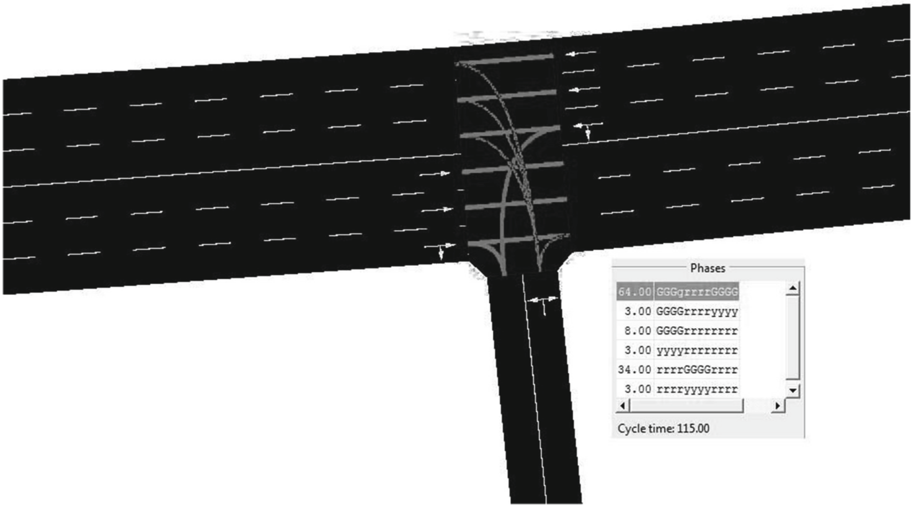

Fig. 1. Phase distribution

the red light. NV is the number of cars that have not reached their destination during the simulation. The denominator is the sum of squared number V of vehicles that have reached their destination within a simulation run and an additional parameter P, standing for phase balance. This parameter is defined as follows

$$P = \sum_{k=0}^{tl} \sum_{j=0}^{ph} s_{k,j} \frac{G_{k,j}}{r_{k,j}},$$

where $G_{k,j}$ is the number of traffic lights in green and $r_{k,j}$ is the number of traffic lights in red in the phase state j of duration $s_{k,j}$ on the intersection k.

3 PSO Based Heuristic

Particle Swarm Heuristic firmly took its place in the list of the most simple and at the same time effective evolutionary based approaches. It simultaneously combines the advantages of the trajectory based approaches together with pluses of the evolutionary methods. It was first proposed in [9] as a concept for optimization of nonlinear functions. The idea of the approach was inspired by the behaviour of the organisms in a bird flock or fish school. Being initially designed for continuous optimization problems nowadays PSO is efficiently applied to the discrete problems as well. It is a population-based iterative approach. In each step of the algorithm a number of particles (represent solutions) form a population, a swarm. Each particle corresponds to an encoded solution x_i. In each iteration k each particle i updates the corresponding solution x_i according to its velocity of movement v_i:

$$x_i := x_i + v_i.$$

The velocity and direction of movement of each particle is guided by its own best known position as well as the best known position of the whole swarm:

$$v_i := \omega v_i + \phi_p U(0,1)(p_i - x_i) + \phi_b U(0,1)(b - x_i).$$

Here p_i denotes the best known position of particle i during the history of the search, b denotes the best known position of the whole swarm. Coefficients ω, ϕ_p and ϕ_b corresponds to the inertia force, and an impact of the personal and the swarm best position on the direction of movement. As a matter of fact these coefficients represent the only possible tuning parameters in PSO. $U(0,1)$ represents a random number drawn from the uniform distribution over the open interval $(0,1)$, independently for each particle in each iteration. The pseudocode of the whole approach can be presented as follows:

Algorithm 1. PSO

Initialization. Generate particles x_i, their initial velocity v_i.
Evaluate particles, calculate $f(x_i)$. Put $p_i = x_i$, $b = p_m, m = \arg\max_i(p_i)$
while a termination criterion is not met **do**:
    for each particle $i = 1, ..., S$ **do**
        Pick random numbers: r_p, r_b from $U(0,1)$
        Update the particle's velocity:$v_i := \omega v_i + \phi_p r_p (p_i - x_i) + \phi_b r_b (b - x_i)$
        Update the particle's position: $x_i := x_i + v_i$
        Evaluate particles, calculate $f(x_i)$
        if $f(x_i) < f(p_i)$ **then**
            update the particle's best known position $p_i := x_i$
        end if
        if $f(x_i) < f(b)$ **then**
            update the swarm's best known position $b := x_i$
        end if
    end for
end while
Return b as best solution found

Algorithm 1 describes the pseudo-code of PSO. The algorithm starts by initializing the swarm. Each component of each particle is generated at random using a uniform distribution over a predefined interval. The same is then done with the velocities. Then all particles are evaluated via a simulation run. Then the main cycle is started. During a predefined number of iterations the following cycle is processed: particles velocities are updated and each particles position is updated according to its velocity. There are three forces that drive each particle - its own inertia weight which draws the particle in the same direction, traction for the best position of this particle and for the best position found by all particles in the swarm. The values of the coefficients ω, ϕ_p and ϕ_b should be chosen accordingly. Higher values of ω correspond to the exploration search, while lower values lead to exploitation of the promising region. The balance between ϕ_p and ϕ_b defines the advantage of individual solution over the general one. Optimal values of the parameters can be found during a meta-optimization process, or optimized during the search process. After the movement each particle is evaluated during the simulation run. Then the best known positions of particles and the whole swarm are updated and the process repeats. As no convergence is guaranteed the algorithm terminates after a predefined number of steps. The best particle found so far is taken as an answer.

The idea of a union between heuristics and simulation modelling is not new. But, when dealing with complex systems the simulation run becomes an expensive procedure which requires a lot of computational time. While simulation itself is unavoidable being the only way to estimate the solution quality, it can be used more efficiently. In this work we propose the following local search procedure aimed at improving the outcome from a simulation. Among other options, SUMO allows installing virtual detecting loops on selected sections of road network. These loops serves as detectors and accumulate traffic data on the segment of the road. The quantity of such loops does not affect the simulation runtime, so one is able to use this feature during every run of SUMO. Being placed before and after the intersection these detectors can provide a precise information on the number of cars, that have passed the crossing on green or stopped on red. This information, collected from all the directions of the intersection is then used to improve the schedule of a particular TL. If we observe that one direction is overcrowded and the queue on the stop line is only growing, we redistribute the green time in favor of this direction in a predefined proportion. The schema of this improvement for Energetikov roundabout can be presented as follows.

Algorithm 2. Local improvement

Initialization. Collect data on the vehicles flows from all directions Q_1, Q_2, Q_3.
if $Q_1 \geq Q_2 + Q_3$ **then** add 2s to the 1_{st} phase
    for $i \leq 2$ **do**
        subtract 1sec from the 4_{th} phase with probability $p = Q_2/(Q_2 + Q_3)$,
      subtract 1sec from the 3_{rd} phase otherwise
    end for
end if
if $Q_3 \geq Q_1 + Q_2$ **then** subtract 2s from the 1_{st} phase
    for $i \leq 2$ **do**
        add 1sec to the 3_{rd} phase with probability $p = Q_1/(Q_1 + Q_2)$,
      add 1sec to the 4_{th} phase otherwise
    end for
end if
if $Q_2 \geq Q_1 + Q_3$ **then** subtract 2s from the 4_{th} phase
    for $i \leq 2$ **do**
        add 1sec to the 1_{st} phase with probability $p = Q_1/(Q_1 + Q_3)$,
      add 1sec to the 3_{rd} phase otherwise
    end for
end if

4 Numerical Experiments

The proposed approach was implemented in Python environment and tested on a real data instances. We considered the Stancionnaya street and Energeticov roundabout, city of Novosibirsk, Russia, for our setting. The Softline company provided us with the measurements of the real traffic flow on this road network, which is known to be one of the most congestioned part of the city. Together with the traffic flow we have obtained a real-life schedule for all of the TL objects on the segment under consideration. On the basis of this data a set of instances was created, representing different times of the day - morning rush hour, evening jams and mid-day traffic. We considered a 30-min simulation settings. The values of the total traffic flow for a usual Friday on the considered segment are respectively 3820, 3660 and 1672 vehicles. Initial simulation runs showed that while in small to normal traffic conditions the real static schedule performs mostly satisfactory it is not the case for the morning and evening rush hours. Also, we noted that the most overloaded part of the considered network is the Energetikov roundabout. Despite its size, this TL regulated junction is unable to carry its functions during rush hours under the current TL schedule. Figure 2 represents the morning jams on this roundabout.

Fig. 2. Morning congestion on the Energetikov roundabout.

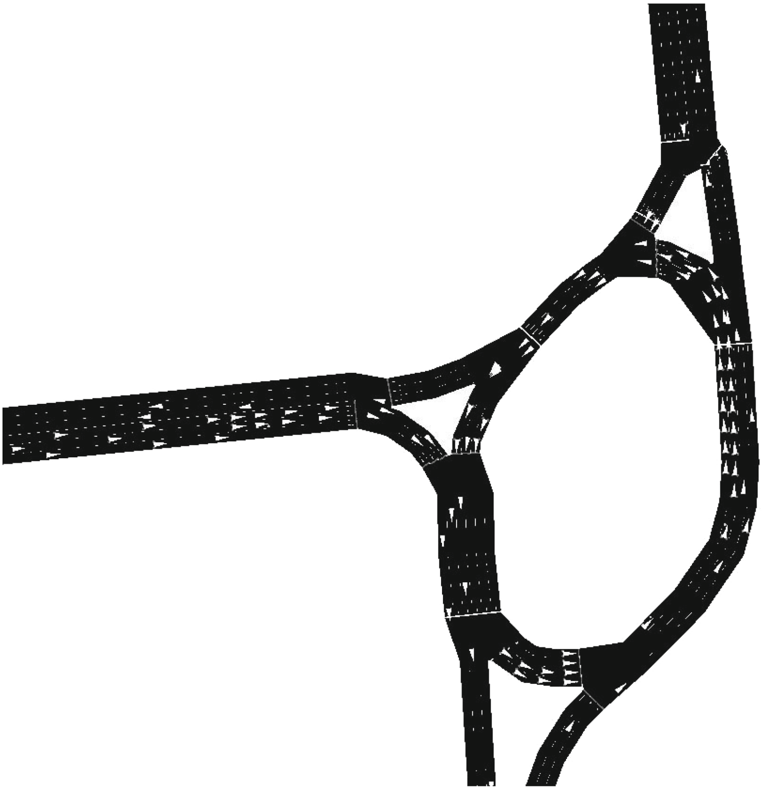

Fig. 3. Optimized scheduling reduces the congestion.

During the first computational experiment we considered only the roundabout itself in a setting of the morning rush hour. For this setting the value of the fitness function with real TL schedule equals 1328 with 2166 vehicles being able to finish their route during the simulation run. We performed 10 runs of our approach, each starting from a randomly generated solution (schedule) and obtained the following results. 9 out of 10 runs converged to the same solution with the value of the fitness function of 964, which gives a 27% improvement. The number of vehicles that arrived to their destination also increased by 13% and reached 2452 vehicles. Similar improvement was also achieved for other instances. The fitness function for the evening hours gained 23%, mid-day case gained 11% rise. Figure 3 demonstrates the resulting improvement. The screenshot is taken in the same moment of the simulation as in Fig. 2. We observe no queue from the south direction and a reduced queue from the west direction.

In the second computational experiment we have considered the whole Stancionnaya str. together with Energeticov roundabout. The results showed that a total improvement of the fitness function value, reached on this segment is close to the results achieved for an isolated roundabout. Although the street itself contains more than 20 crossings and 11 TL objects, the most challenging roundabout junction still remains its bottleneck.

5 Conclusion

We have considered a traffic lights optimization problem. In this work we have proposed an optimization approach, based on Particle Swarm optimization technique in combination with SUMO microsimulation environment. We have tested the proposed approach on an extensive network in Novosibirsk city, Russia. The results shows that the traffic lights schedules provided by our approach outperforms the existing ones and allows to improve the overall quality of traffic in the city. The results of this study may further be used in the planning and constructing of VANET networks.

References

1. McKenney, D., White, T.: Distributed and adaptive traffic signal control within a realistic traffic simulation. Eng. Appl. Artif. Intell. **26**(1), 574–583 (2013)
2. Zheng, X., Recker, W.: An adaptive control algorithm for traffic-actuated signals. Transp. Res. Part C: Emerg. Technol. **30**, 93–115 (2013)
3. Jovanovic, A., Nikoli, M., Teodorovic, D.: Area-wide urban traffic control: a bee colony optimization approach. Transp. Res. Part C: Emerg. Technol. **77**, 329–350 (2017)
4. Li, J.: Discretization modeling, integer programming formulations and dynamic programming algorithms for robust traffic signal timing. Transp. Res. Part C: Emerg. Technol. **19**(4), 708–719 (2011)
5. Coogan, S., Kim, E., Gomes., G., Arcak, M., Varaiya, P.: Offset optimization in signalized traffic networks via semidefinite relaxation. Transp. Res. Part B: Methodol. **100**, 82–92 (2017)
6. Gao, K., Zhang, Y., Sadollah, A., Su, R.: Optimizing urban traffic light scheduling problem using harmony search with ensemble of local search. Appl. Soft Comput. **48**, 359–372 (2016)
7. Garcia-Nieto, J., Alba, E., Carolina Olivera, A.: Swarm intelligence for traffic light scheduling: application to real urban areas. Eng. Appl. Artif. Intell. **25**(2), 274–283 (2013)
8. Transportation Research Record: Highway capacity manual. Technical report, Transportation Research Record (2000)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
10. Krajzewicz, D., Bonert, M., Wagner, P.: The open source traffic simulation package SUMO. In: RoboCup 2006 Infrastructure Simulation Competition (2006)
11. Souravlias, D., Luquey, G., Albay, E., Parsopoulos, K.E.: Smart traffic lights: a first parallel computing approach. In: 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), Ostrawva, pp. 229–236 (2016)

12. Zhang, L., Song, Z., Tang, X., Wang, D.: Signal coordination models for long arterials and grid networks. Transp. Res. Part C: Emerg. Technol. **71**, 215–230 (2016)
13. Angulo, E., Romero, F.P., Garcia, R., Serrano-Guerrero, J., Olivas, J.A.: An adaptive approach to enhanced traffic signal optimization by using soft-computing techniques. Expert Syst. Appl. **38**(3), 2235–2247 (2011)