An Intersection Traffic Signal Controller Optimized by a Genetic Algorithm

Nator Junior Carvalho da Costa and José E. B. Maia Ciência da Computação - CCT Universidade Estadual do Ceará - UECE 60714-903 - Fortaleza - Ceará, Brasil

ABSTRACT

This work addresses the design of actuated traffic controllers, optimized by Genetic Algorithm (GA). The type of sensing used in traffic lanes is a decisive feature for the practical applicability of these controllers due to their complexity and cost, and those based on the measurement of the queue length are the most effective controllers. Sensing the queue length is complex and image-based sensing is typically suggested. The distinguishing feature of this project is that it is based on a binary presence sensor, so technology as simple as an inductive loop can be used. The performance of the controller is evaluated by simulation and the results show that there is only a tolerable reduction in performance when compared to controllers that take queue lengths as inputs.

Keywords

Intersection traffic controller, Traffic simulation, Genetic algorithm, Binary presence sensor

1. INTRODUCTION

The massive volume of vehicles circulating in large metropolitan regions results in long traffic congestion with undesirable consequences for the economy and the quality of life of citizens and with an impact on the natural environment via environmental pollution. The design of traffic controllers optimized to reduce congestion and its consequences is hot research with a lot of recent work being published [2, 5, 9].

The optimization of the controller can be faced directly by the designer by trial and error or by applying techniques such as the design of experiments. This approach, while yielding good results, is time-consuming and costly as the designer is subject to the repetition of this manual procedure each time a change in traffic dynamics is encountered. Therefore, techniques to automate the design of traffic controllers are valuable and their search justified.

Genetic Algorithm (GA) is an effective metaheuristic in search for optimization in high-dimensional spaces with recent applications in various application domains: in engineering project configuration [13], in the design of neural network architectures [14], scheduling project [8], combinatorial optimization [6], continuous controller optimization [7], image processing [12], as well as optimization of traffic controllers [16, 1, 3].

Controlling traffic signals at an intersection is equivalent to defining the sequence and duration of the phases. Phases are each of the ways to allow vehicles to cross the intersection. At the end of each phase, the controller must decide the next phase and its duration.

In general, traffic controllers are classified into three types based on the information and the decision mode used by the controller [19]. Fixed time controllers use a sequence and duration of the preset phases, calculated as a function of traffic regularities. The actuated controllers regulate the sequence and duration of the phases based on detectors located on the network's traffic paths. Finally, adaptive traffic controllers include plan selection or cooperation between local controllers to optimize the individual or collective performance of a group of intersection controllers.

This work addresses the design of actuated traffic controllers, optimized by the genetic algorithm (GA). GAs are an alternative to search-based optimization that follows the gradient, and they are suitable for non-convex optimization. GAs are known for their ability to not get stuck in local minimums.

The type of sensing used in traffic lanes is a decisive feature for the practical applicability of these controllers due to their complexity and cost, and those based on the measurement of the queue length are the most effective controllers. Sensing the queue length is complex and image-based sensing is typically suggested [15]. The distinguishing feature of this project is that it is based on a binary presence sensor, so technology as simple as an inductive loop can be used. The binary sensing adopted here indicates only if the length of the queue is greater than or equal to a certain threshold. The performance of the controller is evaluated by simulation and the results show that there is only a small reduction in performance when compared to controllers that take queue lengths as inputs.

In Section 2 the research context is presented and some references directly related to it are reviewed and in Section 3 the methods are detailed. Section 4 is about the experimental plan, simulations and results. Section 5 concludes by highlighting the main findings of the research.

2. METHODS

This section describes the structure of the traffic controller and the interaction between the GA, the controller and the intersection traffic simulator in the optimization process. The details of GA are also presented.

2.1 Controller Structure

In many optimized traffic controller designs, the operation of the controller takes the queue lengths at the intersection as inputs. Sensing the queue lengths is complex and expensive when compared to binary presence sensors such as inductive loops. Thus, the distinctive feature of the project in this work is that it is a controller based on a binary detector with satisfactory performance when compared to an optimized controller based on measuring queue lengths. Figure 2 shows the location of presence sensors indicating when the length of each queue is greater than or equal to a certain threshold th. By hypothesis, if the sensor th is on, the vehicle queue length is greater than or equal to th.

The three main variables to be decided in the design of this trafficdriven controller are the sequence of the phases, the green times of each phase and a queue length threshold (th) to be detected by the presence sensor. In this project the sequence of the phases is fixed and the green time of the next phase is decided at the end of the previous phase and depends on the reading of the binary presence sensors.

A presence sensor is installed on each traffic lane at a distance from the intersection such that the active sensor indicates that the queue length is greater than or equal to th cars and the inactive sensor indicates that the queue length is less than th cars. In this way, the traffic controller structure is defined and the green times and threshold th will be optimized by a genetic algorithm.

For the signals of the intersection of four phases that will be used in the performance tests of Section 3, each reading of the presence sensors generates one of the sixteen binary vectors [0,0,0,0]to [1,1,1,1].

2.2 The Genetic Algorithm

The interaction between the Genetic Algorithm, the Traffic Controller and the intersection traffic simulator is shown in Figure 1. The general operation is that the fitness assessment of each individual in a GA population requires a complete run of the simulator. Also note that, to perform the performance comparison, the controller input can be binary detectors or queue lengths.



Fig. 1. Functional block diagram of the interactions between the GA, the traffic controller and the intersection traffic simulator during the optimization procedure.

A standard Integer-Coded Genetic Algorithm [4] was used, the pseudo-code of which is shown in Algorithm 1. The next paragraphs describe the constituent parts of the GA used.

Evaluation function: Each chromosome represents a traffic controller configuration. Thus, each individual in a population is eval-

Algorithm 1: Pseudo-code of a standard genetic algorithm.

 Input: crossover probability pc, mutation probability pm

 Output: $best(\mathbf{P})$

 initialize \mathbf{P}
 $\mathbf{P} \leftarrow eval(\mathbf{P})$
 $\mathbf{P} \leftarrow eval(\mathbf{P})$
 $\mathbf{P} \leftarrow select(\mathbf{P}, \mathbf{f})$
 $t \leftarrow 1$

 while $not_stopping_criterion$ do

 $\mathbf{P} \leftarrow reproduce(\mathbf{P}, \mathbf{f}, pc)$
 $\mathbf{P} \leftarrow variate(\mathbf{P}, pm)$
 $\mathbf{f} \leftarrow eval(\mathbf{P})$
 $\mathbf{P} \leftarrow select(\mathbf{P}, \mathbf{f})$
 $t \leftarrow t + 1$

 end

 return $best(\mathbf{P})$

uated through a simulation run, with GA fitness being the average queue delay obtained in that simulation run.

Chromosome representation: GA uses constrained-integer-coded representation for the chromosome. Each gene encodes the green time in seconds for a {phase, input-vector} combination. Therefore, each chromosome is an integer vector with 17 elements as can be seen by the following calculation: 16 inputs + 1 threshold (*th*). Each green time gene is an integer in the range [7,55] [17] and the threshold gene is an integer in the range [1,15]. Thus, in this project a genotype is mapped into a phenotype without transformation.

Reproduce: This procedure has two stages - selection for reproduction and crossover. The selection of parents for reproduction used tournament selection which selects the best individuals among the chosen k. The crossover operation selects genes from parent chromosomes and creates a new offspring. Single point crossover with probability of crossover pc was used.

Variate: This procedure corresponds to mutation. Each chromosome gene is chosen for mutation with probability *pm*. The mutation consists of replacing the current gene with another integer generated randomly in the gene's range.

Select: This procedure is the selection of the next generation. Here the roulette wheel method with elitism was used in which the two best chromosomes are kept for the next generation. The population is completed by selecting individuals with probabilities proportional to their fitness.

Table 1 shows the other GA parameters.

Table 1. Genetic Algorithm configuration

purumeters.				
parameter	value			
Population size	40			
Number of Tournament Competitors	10			
Crossover Probability (pc)	90%			
Mutation Probability (pm)	10%			
Number of Generations	100			

3. RELATED WORKS

Traffic dynamics at an intersection are non-linear and stochastic, and although queue length is a discrete variable, traffic controllers based on queue length are often designed with continuous entry space. Neural and fuzzy controllers [2] as well as model-based designs [18] are frequent. Formulation as a combinatorial optimization problem [11], in addition to other design techniques for continuous nonlinear controllers, such as that based on linearization [10], are also used.

Research and publication on the design of optimized traffic signal controllers is extensive. To contextualize this paper, this section reviews some works directly related to this research.

The paper [3] focuses on the automatic optimization of the passage of vehicles through intersections. The problem has been addressed by proposing three mechanisms to model any type of intersection, to calculate the roads with fewer points of conflict between their inputs and outputs, and to optimize the arrival rate of vehicles using a Genetic Algorithm to achieve the maximum performance of the intersection. The proposed systems achieve a throughput improvement between 9.21 and 36.98% compared with the traditional solutions (priority and traffic lights).

The paper [9] is aimed at optimizing delay at isolated signalized intersections through application of meta-heuristic search optimization methods. However, real-time traffic is usually heterogeneous, having non-linear, stochastic, and intricate characteristics. Thus, they proposed a couple of meta-heuristic-based including GA and DE (Differential Evolution) methods for efficient traffic control. Both GA and DE yielded rational signal timing plans. The results indicated that both methods effectively reduced the average travel time delay ranging from 15 to 35% compared to existing conditions.

Araghi [2] develops the design of a traffic signal controller based on ANFIS (Adaptive-Network-Based Fuzzy Inference System). The controller takes the length of queues at the end of each phase as input to decide both the next phase and its duration. The controller is optimized using Genetic Algorithm. This paper is the main reference of this research and its results are taken as a benchmark for external comparison. Unlike the controller proposed here, in [2] the phase sequence is not fixed. In Section 4 more will be said about this work.

Common to all of this work is that its controllers are based on measuring queue lengths for decision making by the controller. The work in this paper is based on a binary detector which distinguishes it from the previous ones.

4. EXPERIMENTS AND RESULTS

In order to perform the tests, a simplified traffic simulator driven by discrete time with a time pulse equal to 0.1 s was programmed. The simulator abstracts some physical properties of the phenomenon, but it has been calibrated and validated, reproducing approximate results of the average queuing times of some published works. A vehicle start-up time of 2 s has been included in the model. The simulator works as a state machine with transitions every 0.1 s. The state trajectory log file during the simulation is stored and then used at the end of the run to calculate the performance index statistics. Using the classification given in [20] this simulator is in the mesoscopic traffic simulator category.

4.1 Experiment plan

For the tests, an intersection of four phases was simulated with the configuration shown in Figure 2. The phases are those represented in Figure 3. In any of the traffic lanes the vehicle can move forward

or deviate left or right. In all experiments, traffic is a Poisson arrival process.



Fig. 2. Traffic lanes and queues at the four-phase intersection used in the simulation experiments.

Phases Configurations		
1	West East	*
2	East West	4
3	North South	\Rightarrow
4	South North	\blacklozenge

Fig. 3. The directions of the traffic routes of the phases at the intersection.

The experiments compare the performance of the binary presence detector based controller (bd-ga) against that of the one based on measuring the queue length (ql-ga). Both controllers have the same operating structure as described in Subsection 2.1 and are optimized by the same GA. The comparison is made for balanced traffic (equally divided between phases) and unbalanced traffic, varying the total rate at the intersection. For unbalanced traffic, 70% of total traffic was used in one of the phases and 10% in each of the others.

For external performance comparison the results of the work [2] are taken as a reference. In this case, only the result for balanced traffic between phases with a total rate of 1200 car/s is available. The numerical data for the table and graphs were extracted directly from paper [2].

4.2 Results

The results presented are the average of 30 simulation runs for each parameter setting. Low variance (less than 0.1) was observed in the statistics for 30 runs. For this reason, and considering the gap between the values in Table 4.2, statistical hypothesis tests were dispensed with.



Fig. 4. The best fitness of the population in a GA run. The existence of several suboptimal controllers in the final population was observed.

Figure 4 shows the evolution curve of the best fitness of the population in a typical GA execution. Note that an optimal controller was obtained after 60 generations. It was observed that the optimal controller is never obtained after 80 generations. It was also observed that there are several almost optimal controllers in the final population of GA in each execution.

Figures 5 *up* and *down* show the total queued waiting delay accumulated for each of the controllers during a typical simulation run lasting 3600 s, for balanced and unbalanced traffic, at a total rate of 1200 car/s. This figure shows a clear hierarchy of performance among the controllers.



Fig. 5. **Up**: cumulative queued time, unbalanced traffic, 1200 car/s. **Down**: cumulative queued time, balanced traffic, 1200 car/s.

The performance differences observed in Figure 5 are captured in summary form by the average waiting time in line, shown in Table 2 and in Figures 6 *up* and *down*, when the total traffic and the type of traffic (bal, unbal) are varied. For lack of results in [2] for similar

experiments, the third column of Table 2 contains only one value and the graphs of Figures 6 *up* and *down* compare only the controllers developed in this work with the exception being the 1200 car/s setting balanced in Fig-5 *up*.



Fig. 6. Up: average queue delay, balanced traffic. Down: average queue delay, unbalanced traffic.

Discussion: The results in Figures 5 and 6, and in Table 2, reveal the impact of the type of traffic sensing, measuring queue length or binary presence detector, on the performance of the optimized controller. For controllers with the same proposed structure, the average delay of waiting in line goes from 8.18 s to 10.59 s for balanced traffic, and from 5.96 s to 13.70 s for unbalanced traffic, respectively, for 1200 car/s. However, accepting this increase in the average queuing time is advantageous in view of the reductions in complexity and in the cost of operating traffic control at the intersection. With these delay values, the quality of the traffic control service at the intersection perceived by drivers remains satisfactory [17].

When comparing performance with the ANFIS [2] controller, this one is superior to the proposed controller and also to its variant based on measuring the queue length. This is to be expected since no two restrictions relevant to its practical applicability were imposed in the design of the controller in [2]:

- —The ANFIS controller was not imposed a minimum green time. This means that in order to optimize his performance index he can assign green times as short as 1 s while traffic engineering manuals [17] recommend green times of no less than 7 s in Major Arterial.
- —The ANFIS controller was not required to allocate a guaranteed minimum capacity for all phases. In this condition, an imbalance of traffic can impose long waiting times in queue on low flow traffic routes while seeking to reduce the global average.

With these restrictions, the ANFIS controller is difficult to apply.

Table 2. Average queue delay obtained by				
controllers for balanced traffic and				
unbalanced traffic. For lack of results in [2]				
for similar experiments, the fourth column				
contains only one value				

contains only one value.						
traffic	bd-ga	ql-ga	ANFIS [2]			
bal-800	10.51	7.13	-			
bal-1200	10.59	8.18	4.87			
bal-1600	11.21	9.51	_			
unbal-800	12.16	4.67	-			
unbal-1200	13.70	5.96	-			
unbal-1600	15.23	7.35	-			

5. CONCLUSION

This research explored the abilities of genetic algorithms in the optimization of an actuated signal controller at intersections. The distinguishing feature of this design is that the controller is based on binary detection in contrast to most designs which are based on measuring queue lengths. This feature of binary sensing has a strong impact on the practical applicability of the controller.

The comparative performance tests using simulation found that this change in the detection sensor generates only a small and acceptable reduction in the performance of the controller when compared to measuring queue length, remaining, however, much higher than the pre-timed controllers.

In continuity, this research will seek to apply this design principle to Area Control in which multiple interdependent intersections must be coordinated.

6. REFERENCES

- [1] Afzal Ahmed, Syed Ahsan Ali Naqvi, David Watling, and Dong Ngoduy. Real-time dynamic traffic control based on traffic-state estimation. Transportation research record, 2673(5):584-595, 2019.
- [2] Sahar Araghi, Abbas Khosravi, and Douglas C Creighton. Anfis traffic signal controller for an isolated intersection. In IJCCI (FCTA), pages 175-180, 2014.
- [3] Luis Cruz-Piris, Miguel A Lopez-Carmona, and Ivan Marsa-Maestre. Automated optimization of intersections using a genetic algorithm. IEEE Access, 7:15452-15468, 2019.
- [4] Leandro Nunes De Castro. Fundamentals of natural computing: basic concepts, algorithms, and applications. CRC Press, 2006.
- [5] Wade Genders and Saiedeh Razavi. An open-source framework for adaptive traffic signal control. arXiv preprint arXiv:1909.00395, 2019.
- [6] Tintu George and T Amudha. Genetic algorithm based multiobjective optimization framework to solve traveling salesman problem. In Advances in Computing and Intelligent Systems, pages 141-151. Springer, 2020.
- [7] Parul Goyal, Hasmat Malik, and Rajneesh Sharma. Application of evolutionary reinforcement learning (erl) approach in control domain: A review. In Smart Innovations in Communication and Computational Sciences, pages 273-288. Springer, 2019.

- [8] Ali Asghar Rahmani Hosseinabadi, Javad Vahidi, Behzad Saemi, Arun Kumar Sangaiah, and Mohamed Elhoseny. Extended genetic algorithm for solving open-shop scheduling problem. Soft computing, 23(13):5099-5116, 2019.
- Arshad Jamal, Muhammad Tauhidur Rahman, Hassan M [9] Al-Ahmadi, Irfan M Ullah, and Muhammad Zahid. Intelligent intersection control for delay optimization: Using metaheuristic search algorithms. Sustainability, 12(5):1896, 2020.
- [10] Marcella P Lazar and JE Bessa Maia. Feedback linearization controllers for rov depth control. In CONFERÊNCIA BRSILEIRA DE DINÂMICA, CONTROLE E APLICACÕES (DINCON 2017), 2017.
- [11] Shu Lin, Bart De Schutter, Yugeng Xi, and Hans Hellendoorn. Fast model predictive control for urban road networks via milp. IEEE Transactions on Intelligent Transportation Systems, 12(3):846-856, 2011.
- [12] Seyedali Mirjalili, Jin Song Dong, Ali Safa Sadiq, and Hossam Faris. Genetic algorithm: Theory, literature review, and application in image reconstruction. In Nature-Inspired Optimizers, pages 69-85. Springer, 2020.
- [13] Amir Hosein Habibi Onsorodi, Orhan Korhan, et al. Application of a genetic algorithm to the keyboard layout problem. PloS one, 15(1):e0226611, 2020.
- [14] Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Jiancheng Lv. Automatically designing cnn architectures using the genetic algorithm for image classification. IEEE Transactions on Cybernetics, 2020.
- [15] Anuradha G Suratekar and Uttam L Bombale. Wsn based cost effective intelligent traffic light control system based on image processing. International journal of computer applications, 132(7), 2015.
- [16] Dusan Teodorovic, Panta Lucic, Jovan Popovic, Shinya Kikuchi, and Branimir Stanic. Intelligent isolated intersection. In 10th IEEE International Conference on Fuzzy Systems.(Cat. No. 01CH37297), volume 1, pages 276-279. IEEE, 2001.
- [17] Brian Wolshon, Anurag Pande, et al. Traffic engineering handbook. John Wiley & Sons, 2016.
- [18] Ahmet Yazici, Gangdo Seo, and Umit Ozguner. A model predictive control approach for decentralized traffic signal control. IFAC Proceedings Volumes, 41(2):13058-13063, 2008.
- [19] Dongbin Zhao, Yujie Dai, and Zhen Zhang. Computational intelligence in urban traffic signal control: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4):485-494, 2011.
- [20] Xuesong Zhou and Jeffrey Taylor. Dtalite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration. Cogent Engineering, 1(1):961345, 2014.