Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Genetic algorithm-based traffic lights timing optimization and routes definition using Petri net model of urban traffic flow

Henrique Dezani*, Norian Marranghello, and Furio Damiani*****

* School of Technology of São José do Rio Preto, Brazil (Tel: 55 17 991354070; e-mail: dezani@fatecriopreto.edu.br)
**Department of Computer Science, São Paulo State University, Brazil, (e-mail: norian@ibilce.unesp.br)
***School of Electrical and Computer Engineering, University of Campinas, Brazil (e-mail: furio@dsif.fee.unicamp.br)

Abstract: This work presents an algorithm for the optimization of urban traffic flow that computes the vehicle routes and traffic lights timing in real time. The optimization procedure uses a genetic algorithm whose fitness function consists of a high-level Petri net model of the urban traffic flow, which simulation results in the fitness value to be used. The outcome of this work is the optimization of urban traffic flow by simultaneously establishing the best possible routes for each vehicle and the definition of the most appropriate traffic lights timing. According to the tests, the simultaneous optimization of traffic lights time and vehicle routes decreased the total travel time as compared to the optimizations performed considering only the routes.

1. INTRODUCTION

Everyday more humankind suffers with the effects of the ever increasing number of vehicles flowing through the urban roads. Such a steadily augmenting flow affects, for instance, the environment, by the high pollutant emission to the atmosphere, the society, due to the large amount of time wasted in traffic jams, and the economy, for the high financial cost of providing the required infrastructure. According to the World Commission on Environment and Development (1987), sustainable development involves the fulfillment of present generation needs without compromising the capability to supply future generation necessities. Therefore, all research activity engaged in solving the traffic flow problem is a serious candidate to benefit humankind.

Intelligent Transportation Systems (ITS) researches focus on the sustainability concept. They consist of the integration of systems that use both emergent and established technologies in order to monitor and control the transportation systems, to operate them in an efficient and secure way (Ni, 2007).

ITS should include means to obtain vehicle positioning data and to allow access to these data in order to produce useful information for the drivers. Such information would include both faster as well as jammed routes indications. A research area with this goal is the investigation of the communication architecture among vehicles and infrastructure (V2I), which enables vehicles to send data, such as current velocity and position, to a central server (Miller, 2008).

Among Intelligent Transportation Systems, there are Advanced Traveler Information Systems (ATIS), which use a variety of technologies, such as Internet, mobile phones, radios, etc., to help drivers make decisions on their trajectories. The information provided by these systems generally consists of optimized routes or accident/incident notifications that could disturb the trip of the drivers. This data can be accessed before or during the trips (Kumar *et al*., 2005).

Regarding the urban traffic structure, we can consider that this is event-driven and asynchronous, and its evolution in time depends on the interactions of several discrete events, such as arrival and departure of vehicles at a particular intersection and the complete accomplishment of a traffic light cycle. Considering the urban traffic as a discrete event system, the use of Petri nets (Murata, 1989) to model it is interesting for its parallel and asynchronous characteristics. There are several works that used Petri nets and its extensions in the modeling of intersections and traffic lights. In the work of Di Febbraro *et al.* (2004) is presented a model of timed Petri net in which the tokens are the vehicles and the signs are the roads or intersections. In the paper of Dotoli and Fanti (2006) is proposed a model of Coloured Petri net (Jensen and Kristensen, 2009) that describes the urban traffic flow from a microscopic view, in which the tokens represent the vehicle path and the places are the intersections or roads. In the work of Vàzquez *et al.* (2010) is presented a model of a single intersection using Hybrid Petri nets, which roads are represented by places with continuous values. In this same work, traffic light intervals in the intersection are represented by places with discrete values, and the vehicle routes are unknown.

The main advantage of using Petri nets is that they have a support for the analysis of various properties and problems related to systems with concurrency. Reachability is used for analysis of behavioral properties and is a fundamental method for studying the dynamic properties of any system, where the firing of an enabled transition will change the token distribution in a net (the marking of the net).

Even with the increasing number of vehicles traveling in urban roads, these pathways are not properly adapted to meet such a demand, quite often due to the lack of space to enlarge them. On the other hand, it can be verified that some roads are not frequently utilized, either for the drivers' lack of habit in using them or for the larger travel duration indicated by present day positioning systems. Ghali and Smith (1995) argued that the simple control of urban traffic in some roads is not enough to avoid traffic jams. They advocate that the awareness of the drivers must be provoked to motivate them to use other routes.

From this premise and assuming real time vehicle communication and infrastructure (V2I) we propose a system aiming at optimizing vehicle flow in urban roads. The optimization is achieved through the adjustment of two parameters, namely: route definition and green interval setting. The best route for a vehicle is determined on the basis of the current routes and positions of all the vehicles traveling through the urban roads considered. At the same time, the green time intervals for the traffic lights are computed and reset wherever applicable.

This paper is organized as follows: in section 2 an overview of urban traffic is presented; in section 3 the vehicles traffic Petri net model is introduced together with the proposed genetic algorithm; in section 4 the tests conducted and the results obtained are discussed; and in section 5 some conclusions are given.

2. URBAN TRAFFIC STREAM OVERVIEW

Traffic stream is made up of individual drivers and vehicles interacting with each other and the physical elements of the roadway and its general environment. In an urban center the traffic flow is interrupted, since it incorporates external interruptions into their operation, e.g., the red intervals of traffic lights. These interruptions create queues when the signal is on red interval and, if there is no congestion, these queues dissipate during the green interval (Roess *et al.*, 2010).

The problem of determining the green traffic light interval for constant vehicle flow is not easy to solve since there are several intersections related with different configurations. For a feasible solution in a timely manner, systems use heuristics to solve this problem. One of the programs to determine traffic light intervals is TRANSYT (Wong et al., 2002). It's a computer program that operates off-line on a fixed time interval. It is used either for designing, modeling or studying intersections, which can be either isolated or connected to the urban traffic network. The optimization techniques used in TRANSYT are both Hill Climbing and Simulated Annealing. Another program is SCOOT (Split Cycle and Offset Optimization Technique) (Robertson and Bretherton, 1991), an ATCS (Adaptive Traffic Control System) which monitors real-time traffic flow and makes small adjustments to the traffic light timing in order to decrease the vehicle waiting time. Both systems use traffic flow prediction in their optimization program.

Determining time intervals of traffic signals based on the number of vehicles on roads, without its route information, makes it difficult to reduce the waiting time for all vehicles that follow the same route given by GPS. In general, GPS systems use some algorithm to define the best route based on time or distance of travel, e.g., Dijkstra shortest path (DongKai and Ping, 2010) and, for the same origin and the same destination, the route is always the same.

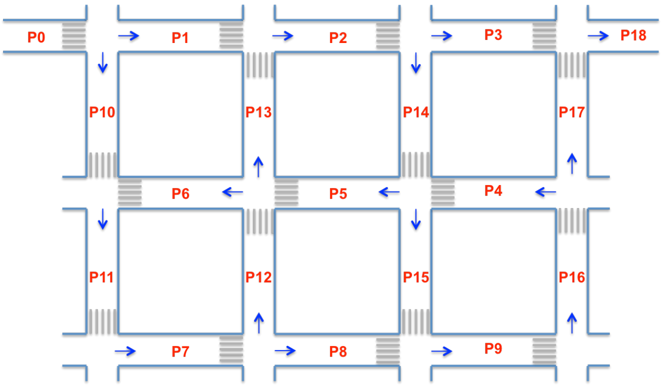

Fig. 1. Controlled urban roads

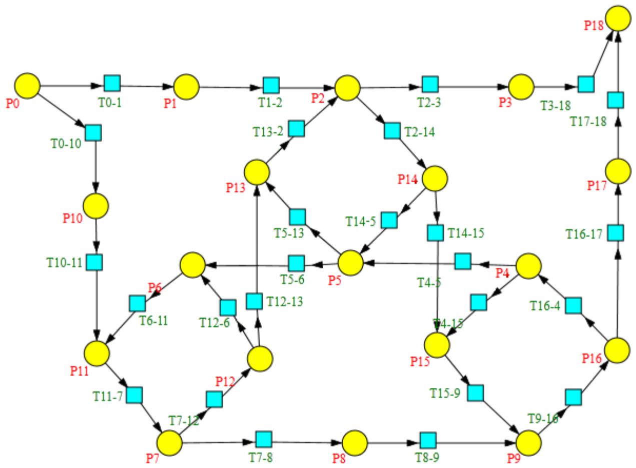

Fig. 2. Petri net urban traffic model

The routes indicated by the GPS systems could be the shortest, but if there are a lot of vehicles and all these follow the same route, there will be necessary to wait too long for these vehicles on some roads. Using the other works that used Petri net described here, the route of the others vehicles routes also are not considered and so, the shortest path found by them can not represent the faster, also causing congestion on some roads. However, if alternative routes are given considering the possible congestion in the roads, even traveling a distance greater, a vehicle may have a shorter time to arrive at its destination.

In a previous paper (Dezani *et al.*, 2014) we presented a genetic algorithm to determine routes for the vehicles traveling on an urban road system. The fitness function used was based on the Petri net model of the traffic system. In the present paper the urban traffic flow optimization is extended to consider not only the best possible routes for the vehicles, but also determining in real time the green time interval for the traffic lights present in the intersections of the considered routes.

3. PETRI NET MODELING OF THE URBAN TRAFFIC

The High Level Petri nets are extensions of original Petri nets (Murata, 1989) proposed by Carl Adam Petri in 1962. These extensions allow the Petri net model to use individual tokens, which carry information that can be used on deciding when to fire transitions. In our work, the information of tokens consists of allowed time for their consumption and a list with their path inside the model. Analogously, the time represents the interval used by a vehicle to cross a road and the path represents the route of the vehicle in a city.

In Figure 1, the urban area controlled by the system is displayed, where each channel has a length of 100 meters. In Figure 2 a High Level Petri net model based on the urban area of Figure 1 is presented. In this model, each place is equivalent to an urban road and may contain from zero to a limited number of tokens, corresponding to the vehicle and the arcs have weights equal to 1. The conflicts generated with the use of boolean tokens are solved when incorporating data types to them. For example, with a token on place *P0*, both transitions *T0*-1 and *T0-10* can be fired, but with the inclusion of the guard transition, where a token has a route to place *P1*, only the transition *T0-1* is fired and if the route is to place *P10*, only transition *T0-10* is fired. Taking into account the execution semantics considered for the High Level Petri net model, following synchronized and/or timed Petri nets (David and Alla, 2001) where the firing of a transition is deterministic dependent on the associated enabling conditions, there is the possibility of firing transitions *T0-1* and *T0-10* in parallel. This occurs only in the case of having two tokens with the same firing time but with different routes. This corresponds to urban roads which have two lines of vehicles.

In Figure 3 it is presented the T-Timed Petri net (David and Alla, 2001) model regarding the traffic light at the intersection given at the entrance of place *P2*, where both transitions *T1-2* and *T13-2* can be fired. For each intersection on the net there is another equivalent model that is not shown for purposes of simplification of the model.

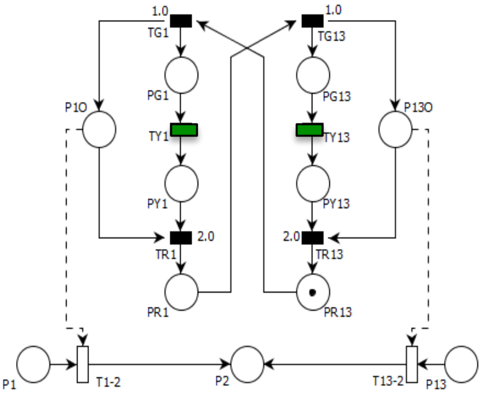

Fig. 3. T-Timed Petri net model of traffic light

Places *PG1*, *PY1* and *PR1* correspond respectively to the green, yellow and red intervals of traffic light *S1*. Places *PG13*, *PY13* and *PR13* correspond respectively to the green, yellow and red intervals of traffic light *S13*. When the traffic light *S1* has a token in place *PG1* or *PY1*, there is also a token in place *P1O*, which will be used for enabling transition *T1-2*. The same occurs for the traffic light *S13* and the transition *T13-2*. It is important to note that the arc that connects *P1O* in *T1-2* and the arc that connects *P13O* in *T13-2* are test arcs and therefore, by firing transitions *T1-2* or *T13-2*, the tokens in places *P1O* and *P13O* are not removed.

Transitions *TG1* and *TG13* are respectively responsible for changing the interval of the traffic lights *S1* and *S13* from red into green, and have an associated time delay of 1 second. This interval is also part of the transition from green interval to red interval opposite to traffic light that will open, however, during this time all traffic lights participants of the intersection have a red interval.

The transitions *TY1* and *TY13* are responsible for changing the green interval to yellow, and are fired according to the time found in our optimization, thus keeping the tokens in the entry places (*PG1* or *PG13*) available through this period. After fired, transitions *TY1* and *TY13* generate a token in place *PY1* and *PY13*, respectively, which indicate that the traffic signals are in yellow interval.

The time transitions *TY1* and *TY13* wait for firing as well as the waiting time for the other transitions associated to the green interval of the traffic lights, is subject of the optimization proposed in this paper and explained in section 4.

At last, the transitions *TR1* and *TR13* are fired 2 seconds after being enabled, consume the tokens from places *PY1* and *PY13*, respectively, and create a token in place *PR1* and another in *PR13*, which indicate that the traffic lights are in red interval. The transitions *TR1* and *TR13* are only fired if there is a token in place *P1O* or *P13O* and, when fired, will remove these tokens, disabling the firing of transitions *T1-2* and *T13-2*, respectively. This interval allows for a vehicle that could not stop at the end of the green interval, cross the intersection safely and its duration is known in literature.

4. GENETIC ALGORITHM

The Genetic Algorithms are search algorithms based on mechanisms of natural selection and genetics. They combine the data structure of the fittest individuals using a random exchange of information, its main goal is to strike a balance between efficiency and effectiveness required for use in different environments. Therefore, Genetic Algorithms are developed theoretically or empirically to provide robust search in complex space of solutions (Goldberg, 1989), such as scheduling systems with a large number of parameters.

According to Figure 4, the chromosome of our Genetic Algorithm consists of an array of size equal to the number of vehicles plus the number of traffic lights in urban traffic network. Each gene related to vehicle's route corresponds to one of the many routes of the given vehicle and each gene related to traffic lights corresponds to the green time interval used for it. Thus, during the selection of individuals and their genetic operations, the positions of the genes are not altered so as to maintain the order of the respective vehicles and traffic lights.

The crossover operation is used to exchange of information from a randomly selected point. As the example shown in Figure 4, assuming that the Genetic Algorithm from the roulette selection mechanism has selected the first two chromosomes randomly generated and the point is equal to 2, we would have the following offsprings O_1 = *{R_0, R_1, R_0, R_2, R_3,..., R_0, 14,...,14}* and O_2 = *{R_1, R_3, R_2, R_0, R_2,..., R_1, 10, ..., 14}*.

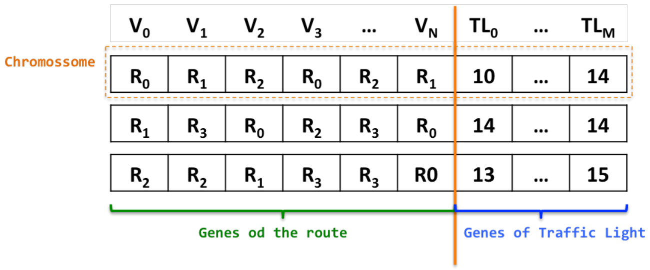

Fig. 4. Chromosome used for the proposed genetic algorithm

Once generated the initial population, each individual of this population is subjected to an evaluation to determine how much favors the optimization of the system. The evaluation is given by the total time spent to dispatch the vehicles using the simulation of the Petri net, which is presented in Dezani *et al*. (2014). The difference between the referenced work and this work is that in the former, time intervals were fixed, while in this work, the times of traffic lights are considered part of the solution, as can be seen in Figure 4.

After computing the fitness for each individual in the population we use the roulette wheel selection to choose the individuals from the population to apply crossover. The parents utilized for the crossover operation are discarded and the offsprings are added to a new population. Once the new generation is completed the process is repeated. This occurs until either all population converges to a winning individual or until 40 generations have been created or a time limit of two seconds is reached. The time limit is related to the maximum time within which the system must return a response to the traffic controller, and depends on the vehicles saturation on the roads and the minimal time used in green interval of the traffic lights. These limits are based on the time set in the work of Dezani *et al*. (2012).

According to the tests, we chose not to apply the mutation genetic operator, since the selection performed is elitist, allowing a faster convergence.

5. MATERIALS ANS METHODS

The tests were performed using a computer with Core i5 2.4 GHz processor and 4GB of RAM memory and an incidence matrix with 25 rows and 31 columns representing the urban traffic model in Petri net presented in Figure 2. It was considered that all pathways have 100 meters between intersections, and each route has a maximum capacity of 10 vehicles at the same time. Lastly, consider that the flow pathways has only one direction as shown by the arrows in Figure 1.

We added one token in place P0 having as destination the place P13. Generating the state space the possible routes were found: R_1 = *{P0, P1, P2, P14, P5, P13}*, R_2 = *{P0, P10, P11, P7, P12, P13}*, R_3 = *{P0, P10, P11, P7, P8, P9, P16, P4, P5, P13}* and R_4 = *{P0, P1, P2, P14, P15, P9, P16, P4, P5, P13}*. Using Dijkstra's shortest path algorithm, we have the lowest route defined by R_D = R_2 = *{P0, P10, P11, P7, P12, P13}*.

Our test consists on uniformly arrival of tokens in place *P0* and in comparing the time taken for all vehicles to reach place *P13*. Every 2 seconds was added a token at place *P0*. It is considered a capacity of 10 vehicles per road.

We described here four cases, which are: case 1 - use only the routes generated by Dijkstra's algorithm and fixed times to traffic lights; case 2 - use only the routes generated by Dijkstra's algorithm and optimization of the times of traffic lights; case 3 - use of the routes generated by our algorithm with fixed time of traffic lights; and case 4 - use of the routes and times of intervals generated by our algorithm.

The Genetic Algorithms utilized in these tests had a population of 2000 individuals and 40 generations, differing only by the amount of available routes for each vehicle. Increasing the population size or the number of generations did not influence the solution provided by the algorithm. However, the decrease in population size, in some cases, returned unsuccessful solutions when the vehicles were associated to a greater number of routes. The time of each traffic light gene was generated using a random function, which produces a number between 10 and 20 seconds. We use this interval based on researches of traffic engineering, which describe not only the time to cross an intersection, but the time spent to start moving the vehicles, called Startup-Lost Time.

6. TESTS AND RESULTS

Figure 5 displays a plot of the amount of vehicles in each road against time. To obtain this plot the same route for all the vehicles as well as a fixed 10 seconds time for the green interval of all the traffic lights were used.

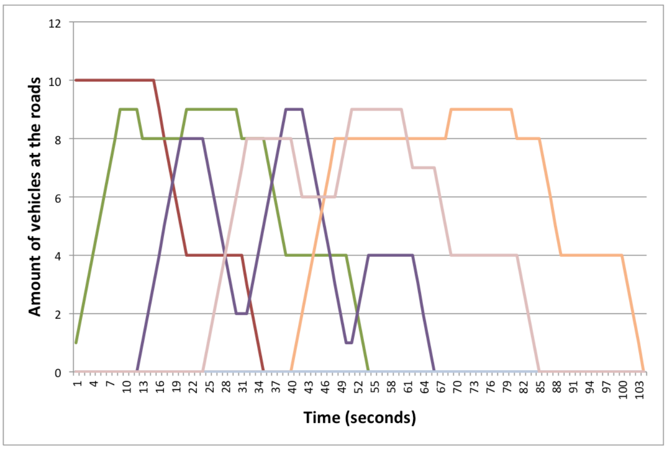

Fig. 5. Plot displaying the amount of vehicles per road over time using the routes determined by Dijsktra's shortest path algorithm and using a fixed green time interval

The total amount of time a set of 20 vehicles required to reach their destinations was 103 seconds. It is important to notice that a large amount of vehicles has been kept on the pathways for a long time frame. This characteristic implies that slower vehicles and accidents will produce a delay in vehicles dispatching. Thus, the smaller the amount of vehicles on the roads or the shorter the amount of time the roads remain at their maximum capacity, the less susceptible to traffic jams the roads will be, even in the eventuality of accidents. Accidents may be notified by the system from the communication infrastructure V2I previously mentioned.

In Figure 6 it is shown a plot of the amount of vehicles over time. Vehicles travelled according to the same routes produced by Dijsktra's algorithm. However, in this case the time for the green intervals was optimized using the genetic algorithm presented in the previous section.

One can observe that the amount of time required for the vehicles to travel to their destination was 10 seconds greater than for the first case. However, the amount of time in which the roads were kept at their maximum capacity was reduced, favoring traffic jams avoidance in the event of accidents. It can also be noticed that at 85 seconds of simulation time there were 8 vehicles for the case presented in Figure 5, while there were only 7 vehicles for the case presented in Figure 6.

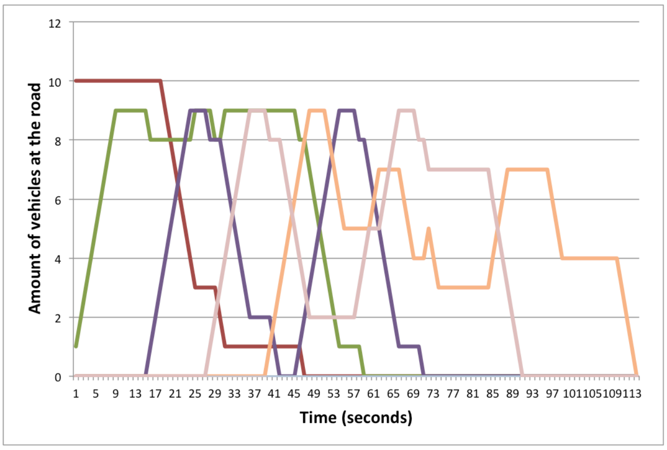

Fig. 6. Plot displaying the amount of vehicles per road over time using the routes determined by Dijsktra's shortest path algorithm and using a variable green time interval

In Figure 7 it is shown a plot of the amount of vehicles in each road over time.

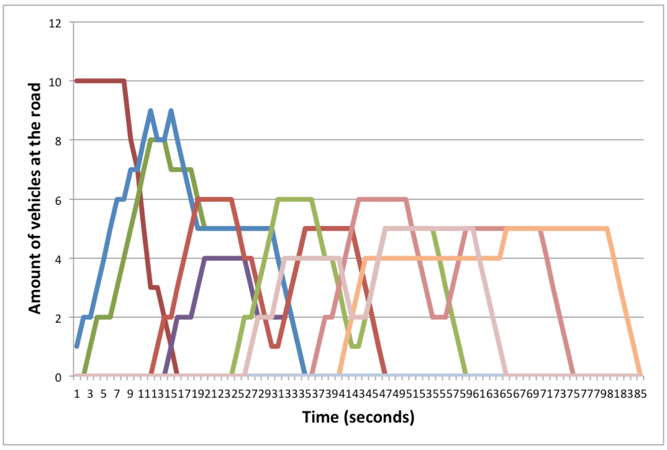

Fig. 7. Plot displaying the amount of vehicles per road (place) over time using the routes determined by the algorithm proposed in this paper and using a fixed green time interval

In the case of the Figure 7 the routes are computed by the algorithm presented in the previous section keeping fixed the green interval times. It can be observed that the urban roads do not reach their maximum allowed capacity, and that roads not used by the Dijkstra's algorithm are now used for the traffic optimization.

Figure 8 displays a plot of the amount of vehicles in each road over time. For this test case a variable green interval has been used for the traffic lights. It can be observed that the urban roads do not use the maximum allowed vehicles capacity.

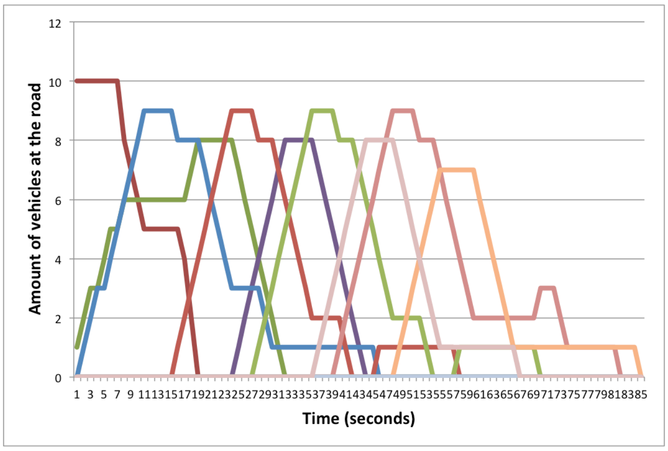

Fig. 8. Plot displaying the amount of vehicles per road over time using the routes determined by the algorithm proposed in this paper and using a variable green time interval

The maximum amount of time required for a set of 20 vehicles to travel to their destinations was reduced in 1 second as compared to the previous case, and in 19 seconds as compared to the first case. Besides, the amount of vehicles in roads *P7*, *P14*, *P12*, and *P5*, respectively represented in green, rose, lilac, and orange, was reduced to a maximum of 3 from the simulation time of 67 seconds and beyond. In turn, for the case shown in Figure 6, with no time optimization, the roads P12 and P5 presented 5 vehicles each. This means that starting at 67 seconds of simulation time, one has 10 vehicles left in travel for the case of fixed time intervals (Figure 7), and only 4 for the case of variable time intervals (Figure 8).

For a total simulation time of 85 seconds the road represented in green in Figure 7, corresponding to place *P7*, retained an amount greater than or equal to 5 vehicles for 16 seconds. On the other hand the same road retained 5 or more vehicles for only 14 seconds. This means that during 36.84% of the time this road was used below the average allowed capacity (this figure drops to 16.27% of the total simulation time). This is much less than the outcomes observed for the case in Figure 6, which assume the values 52.94% and 20.93%, respectively. Such a characteristic can also be observed for the other roads presented in those plots, and entitles us to conclude that simultaneous optimization contributes to keep the vehicles on the roads for the smallest possible time.

7. CONCLUSIONS

With the increasing number of vehicles traveling on urban roads and the emergence of new technologies used for the communication among the vehicles and traffic controls it is possible to create applications that take advantage of these technologies in order to allow an improvement of urban traffic flow and consequently the lives of those needing transportation.

This paper presented an algorithm that utilizes the analyses of High Level Petri net models as a fitness function of one Genetic Algorithm to identify alternative routes in the path of vehicles and the green time interval of the traffic lights within a bounded urban area.

According to tests carried out with the algorithm, it can be said that this application provides a considerable optimization in travel time to all vehicles on urban roads. We conclude that the algorithm optimizes traffic flow in general to use alternative routes, which previously was not being considered as part of the path, at same time that set the green time interval of the traffic lights.

We consider the simulation of routes and travel times since there is no structure capable of obtaining such data for effective practical application. This structure is related to the focus of the Intelligent Transport System studies.

The tests showed that the genetic algorithm created 40 generations to find an optimal result. The processing time of the Genetic Algorithm, defined by parameters explained throughout the text, did not exceed 2 seconds. Therefore, it is fair to say that the proposed algorithm can operate in real time for the class of systems under study.

ACKNOWLEDGMENT

The first author would like to express his gratitude to the CAPES for the financial support.

REFERENCES

David, R. and Alla, H. (2001). On Hybrid Petri Nets. *Discrete Event Dynamic Systems*, **vol. 11**, pp. 9-40.

Dezani, H.; Damiani, F.; Viudes, U.; Parra, I. A. (2012). Mobile application as a tool for urban traffic data collection and generation to Advanced Traveler Information Systems using Wi-Fi networks available in urban centers. *IEEE Intelligent Vehicles Symposium*, p. 288-292.

Dezani, H., Bassi, R. D. S., Marranghello, N., Gomes, L., Damiani, F. and Silva, I. N. (2014). Optimizing urban traffic flow using Genetic Algorithm with Petri net analysis as fitness function. *Neurocomputing*, **vol. 124**, pp. 162–167.

Di Febbraro, A., Giglio, D. and Sacco, N. (2004). Urban traffic control structure based on hybrid Petri nets. *IEEE Transactions on Intelligent Transportation Systems*, **vol. 5**, pp. 224-237.

DongKai, F. and Ping, S. (2010). Improvement of Dijkstra's algorithm and its application in route planning. *Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1901-1904.

Dotoli, M. and Fanti, M. P. (2006). An urban traffic network model via coloured timed Petri nets. *Control Engineering Practice*, **vol. 14**, pp. 1213-1229.

Ghali, M. O. and Smith, M. J. (1995). Managing Traffic Congestion by using Traffic Control. *IEE Coloquium on Urban Congestion Management*, p. 1-8.

Goldberg, D. E. (1989). *Genetic Algorithms in search, Optimization and Machine Learning*, 1st ed., Addison-Wesley.

Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri Nets: Modeling and Validation of Concurrent Systems*. **vol .1**, Springer Verlag.

Kumar, P.; Singh, V. and Reddy, D (2005). Advanced Traveler Information System for Hyderabad City. *IEEE Trans. Intelligent Transportation Systems*, **vol. 6**, pp. 26–37.

Miller, J. (2008). Vehicle-to-Vehicle-to-Infrastructure (V2V2I) Intelligent Transportation System Architecture. *Conf. Rec. 2008 IEEE Intelligent Vehicles Symposium*, pp. 715–720.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, **vol. 77**, pp. 541-580.

Ni, D. (2007) Determining Traffic-flow characteristics by definition for application in ITS. *IEEE Trans. Intelligent Transportation Systems*, **vol. 8**, pp. 181-187.

Robertson, D. I. and Bretherton, R. D. (1991). Optimizing networks of traffic signals in real time-the SCOOT method. *IEEE Transactions on Vehicular Technology*, **vol. 40**, pp. 11-15.

Roess, R. P., Prassas, E. E. and MacShane, W. E. (2010). *Traffic Enginnering*, 4th ed., Prentice Hall.

Vázquez, C. R.; Sutarto, H. Y.; Boel, R. and Silva, M. (2010). Hybrid Petri net model of a traffic intersection in an urban network. *IEEE International Conference on Control Applications (CCA)*, pp. 658-664.

Wong, S. C., Wong, W. T., Leung, C. M. and Tong, C. O. (2002). Group-based optimization of a time-dependent TRANSYT traffic model for area traffic control. *Transportation Research Part B: Methodological*, **vol. 36**, pp. 291-312.

World Comission on Environment and Development (1987) *Our Common Future*. Oxford University Press.