Genetic Algorithm Application for Traffic Light Control

Ayad. M. Turky[1], M.S. Ahmad[1], M.Z.M. Yusoff[1], and N.R. Sabar[2]

[1] Universiti Tenaga Nasional, Km 7, Jalan Kajang-Puchong,
43009 Kajang, Selangor, Malaysia
ayad_b2006@yahoo.com, {sharif,zaliman}@uniten.edu.my
[2] University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

Abstract. In this paper, we describe the design of an intelligent traffic light control based on genetic algorithm. This paper is part of our work in which we attempt to use genetic algorithm in traffic light control and pedestrian crossing. In our approach, we use four sensors; each sensor calculates the vehicle density for each lane. We developed an algorithm to simulate the situation of an isolated intersection (four lanes) based on this technology. We then compare the performance between the genetic algorithm controller and a conventional fixed time controller.

1 Introduction

The monitoring and control of vehicular traffic pose a major challenge in many countries. The escalating number of vehicles in the cities not only has a huge environmental impact, but also results in loss of lives on the road. This situation demands a comprehensive approach involving a system which coordinates the traffic controls for smooth flowing vehicles.

In this paper, we address the problem of increased vehicular flow on the road. There is an urgent need to optimize traffic control algorithms to accommodate the increase in vehicles in urban traffic which experience long travel times due to inefficient traffic light controls. Optimal control of traffic lights using sophisticated sensors and intelligent algorithms, especially in four way traffic junction, could be a possible solution.

In a conventional traffic light controller, the traffic lights change at constant cycle times which are clearly not optimal. The preset cycle time regardless of the dynamic traffic load only adds to the problem. We apply the genetic algorithm theory in the traffic control system to provide an intelligent green interval response based on dynamic traffic load inputs, thereby overcoming the inefficiency of the conventional traffic controllers. Such approach resolves the challenges when sensors placed at every lane in a four-way junction control read the density of vehicles to provide inputs for the algorithms.

2 Related Work

Pappis and Mamdani made the first known attempt to use fuzzy logic in traffic light control for a theoretical simulation study of a fuzzy logic controller in an isolated

J. Yang et al. (Eds.): UNISCON 2009, LNBIP 20, pp. 115–120, 2009.

signalized intersection (2+2 lanes, one-way intersection) [1]. They compare their fuzzy method to a delay-minimizing adaptive signal control with optimal cycle time. The fuzzy controller is equal to, or slightly better than, the adaptive method used for comparison.

Other attempts to use fuzzy logic in traffic light controls are made by Tan, Khalid and Yusof [2]; Niittymaki and Kikuchi [3]; Chen, May and Auslander [4]; Choi [5]; and Conde and Pérez [6].

3 Model Design for Traffic Light

We use our genetic algorithm controlled traffic light system on a four-junction two-way lane. We use four sensors; each sensor detects the vehicle density for each lane.The system calculates the green and red light times to be given for vehicles. It also calculates the vehicles queue behind the red light plus the time taken for each vehicle to arrive at its target destination in static and dynamic modes, i.e., if vehicle ID 4 comes from lane A goes to a destination in lane D1, the system calculates the time that it takes to travel from A to D1.

3.1 Variables

The variables for our system include the input, cellular automata, genetic algorithms operating parameters and the output.

The input variables are as follows:

1. Vehicle Density (VD): a measure of vehicles that pass through a green light.
2. Vehicle Queue (VQ): a measure of vehicles density created behind a red light.

The output variables are as follows:

1. Queue of the Vehicles (QV): Number of Vehicles behind the red light per second in static and dynamic modes.
2. The Duration (D) of travel for each vehicle to arrive a the target destination in static and dynamic modes.

3.2 Cellular Automata

One way of designing and simulating (simple) driving rules of cars is by using cellular automata (CA). CA use discrete partially connected cells that can be in a specific state. For example, a road-cell can contain a car or is empty. Local transition rules determine the dynamics of the system and even simple rules can lead to chaotic dynamics [7].

We use cellular automata algorithm in this paper because it allows us to represent significant events that occur during congestions such as traffic standstill, resume motion, return to standstill again, and so on.

3.3 The Model's Algorithms

In our algorithms, we establish the algorithm steps: initialize population, evaluate population, chromosomes selection and chromosomes recombination.

1. Initialize population: Each chromosome contains two genes, the first gene is red time and the other one is green time. We set the chromosomes population to 100. Chromosomes need to be encoded and this is connected to the problem that genetic algorithm is meant to resolve. We use binary encoding to encode the chromosomes. In binary encoding every chromosome is a string of bits 0 or 1 and it gives many possible chromosomes, even with a small number of alleles. See Figure 1 for an example of chromosomes with binary encoding.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

Fig. 1.

2. Evaluate population: This provides a way to rate how each chromosome (candidate solution) solve the problem at hand. It involves decoding the chromosomes into the variable space of the problem and then checking the result of the problem using these parameters. The fitness is then computed from the result.
   Crossover Fraction: With the crossover fraction=0.8, we used two point crossover operation performed on the parent's generation, the result of which is stored in a mean array. In this array, the parent's generation is merged with the children. These steps are repeated until the total number of the crossover operation is half the size of the initialization. We can then say that the crossover operation is completed.
   Mutation Fraction: With the mutation fraction=0.2, we performed this operation on the parent's generation. From the results in the mean array, a random number is generated and the result of comparison between this number and mutation fraction are determined by the occurrence or non-occurrence of mutations. These steps are repeated until the total number of mutation operations is half the size of the initialization. We can then say that the mutation operation is completed.
3. Chromosome selection: The chromosomes are selected for propagation to future populations based upon their fitness. Chromosomes which have high fitness value have a good chance to be chosen for future population. For selection of chromosomes, we use the "Roulette-wheel with probability of selection that is proportional to fitness" based upon the fitness of the chromosomes. See Figure 2.

We determine the fitness function to identify the solutions. The fitness function is calculated based on many parameters (queue, density, green light time and red light time). Our fitness function consists of two parts:

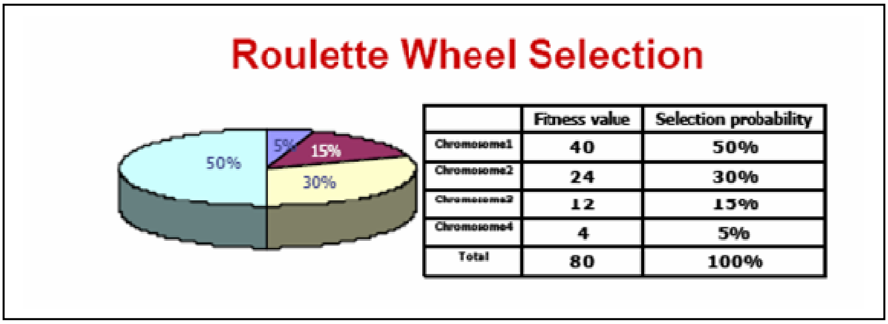

Fig. 2.

a. We calculate the green time necessary due to the queue formed by the red light, `(VQ*vehicle_time_for_passing)` and compare this value with the past green times to obtain a good value for the green time. We set the vehicle_time_for_passing to 3 s.
b. In the same way, we calculate the length of queue, which forms during the red time, `(VD*red)`, and the density of vehicles in the same lane at the same time. The quality of performance increases whenever this value decreases. We give greater attention to optimize the green time at the expense of queue length. Therefore, we multiply a ruling parameter, which change the fitness function as follows:

```
(VD*red)^3-(green-VQ*vehicles_time_for_passing)^2
```

4. Chromosome recombination: In recombination, pairs of chromosomes are recombined, possibly modified, and then placed back into the population as the next generation. The process continues again at evaluation until the problem represented by the chromosomes is solved, or some other exit criterion is met such as convergence, or the maximum number of generations is reached.

The next step in the operation is evaluating the generation to determine the resulting quality of these individuals compared with the previous generation. This is done by arranging the elements of the array (mean array) in increasing values provided by the fitness function.

Ordering the array elements in this way contributes to better identification of individual generations (parent and child generations). The first set of the elements of this array (mean array) is copied to the parent's array. These elements form 70% of the members of the new generation of parents. The rest (30%) is generated by using the random function. The algorithms read the new inputs after five generations to get good solutions.

4 Comparisons between Static and Dynamic Control

We compare the performance between the static control model (fixed cycle time) and our dynamic (genetic algorithm) control model. We use the same inputs for both models. In the static control model, we set the green and the red time for the vehicles to 20 second for each lane. In this control model, the vehicles in one lane will wait even if there is no vehicles queue in the other lane.

In the dynamic control model, the times determined by genetic algorithm depend on the vehicles density (VD) and vehicles queue (VQ) parameters that are read from sensors for each lane resulting in synchronized green and red time. If there are no vehicles queue in one lane, the green time will be zero. Tables 1 and 2 below show the results of our experiments for both modes.

Table 1. Static and Dynamic Queue (Left) and Time Gain (Right)

Time (s)	No. of Vehicles (Behind a Red Light)		V-ID	Duration (s)		Time Gain (s)
	Static	Dynamic		Static	Dynamic	
1	0	0	1	6	5	1
2	0	0	2	16	5	11
3	0	0	3	6	6	0
4	3	2	4	16	7	9
5	4	1	5	16	4	12
6	6	1	6	16	6	10
7	6	1	7	7	8	-1
8	7	2	8	14	10	4
9	8	3	9	12	9	3
10	3	1	10	7	6	1
1	0	0	11	7	6	1

Table 2. Traveling time for Static Mode (Left) and Dynamic Mode (Right)

V-ID	Start Time	Arrival time	Duration (s)	V-ID	Start Time	Arrival time	Duration (s)
1	02:37:12	02:37:18	6	1	02:30:08	02:30:13	5
3	02:37:12	02:37:18	6	3	02:30:08	02:30:14	6
7	02:37:14	02:37:21	7	2	02:30:10	02:30:15	5
10	02:37:18	02:37:25	7	4	02:30:08	02:30:15	7
11	02:37:19	02:37:26	7	5	02:30:12	02:30:16	4
2	02:37:12	02:37:28	16	6	02:30:10	02:30:16	6
4	02:37:12	02:37:28	16	7	02:30:10	02:30:18	8
5	02:37:13	02:37:29	16	10	02:30:14	02:30:20	6
6	02:37:14	02:37:30	16	11	02:30:15	02:30:21	6
9	02:37:18	02:37:30	12	9	02:30:13	02:30:22	9
8	02:37:17	02:37:31	14	8	02:30:13	02:30:23	10

5 Conclusions and Future Work

From the results, we can say that the dynamic control model performs better than the static control model. Due to its flexibility, the dynamic control model is able to calculate the optimal green time based on vehicle density and queue length. Results also show that significant time gain is experienced in traveling through the GA-controlled traffic light system.

In our future work we will extend the application of genetic algorithm in traffic light control systems in which a pedestrian crossing is included. We will address all the likelihood that could happen at a four-way traffic junction for vehicles passing and one lane for pedestrian crossing.

References

[1] Pappis, C.P., Mamdani, E.H.: A Fuzzy Logic Controller for a Traffic Junction. IEEE Transactions Systems, Man, and Cybernetics SMC-7(10), 707–717 (1977)

[2] Tan, K.K., Khalid, M., Yusof, R.: Intelligent Traffic Lights Control by Fuzzy Logic. Malaysian Journal of Computer Science 9(2), 29–35 (1996)

[3] Niittymaki, J., Kikuchi, S.: Application of Fuzzy Logic to the Control of a Pedestrian Crossing Signal. Transportation Research Record: Journal of the Transportation Research Board 1651, 30–38 (1998)

[4] Chen, L.L., May, A.D., Auslander, D.M.: Freeway Ramp Control Using Fuzzy Set Theory for Inexact Reasoning. Transportation Research, Part A 24(1), 15–25 (1990)

[5] Choi, W., Yoon, H., Kim, K., Chung, I., Lee, S.: A traffic light controlling FLC considering. In: Pal, N., Sugeno, M. (eds.) AFSS 2002. LNCS, vol. 2275, pp. 69–75. Springer, Heidelberg (2002)

[6] Conde, C., Pérez, J., González, P., Silva, J., Cabello, E., Monclús, J., Santa Cecilia, T.: A Conflict-Avoiding, Artificial Vision Based, Intelligent Traffic Light Controller

[7] Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Phys. (1-2), 2221–2229 (1992)