2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Novel Traffic Signal Timing Adjustment Strategy Based on Genetic Algorithm

*Hsiao-Yu Tung
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
Email: b98901044@ntu.edu.tw

*Wei-Chiu Ma
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
Email: b98901159@ntu.edu.tw

Tian-Li Yu
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
Email: tianliyu@ntu.edu.tw

***Abstract*—Traffic signal timing optimization problem aims at alleviating traffic congestion and shortening the average traffic time. However, most existing research considered only the information of one or few intersections at a time. Those local optimization methods may experience a decrease in performance when facing large-scale traffic networks. In this paper, we propose a cellular automaton traffic simulation system and conduct tests on two different optimization schemes. We use Genetic Algorithm (GA) for global optimization and Expectation Maximization (EM) as well as car flow for local optimization. Empirical results show that the GA method outperforms the EM method. Then, we use linear regression to learn from the global optimal solution obtained by GA and propose a new adjustment strategy that outperforms recent optimization methods.**

I. INTRODUCTION

Traffic congestion has been a crucial problem in modern cities, especially during rush hours. It not only leads to the increase in travel time people spent but also impacts our environment and economy. According to a latest research [9], the consumption of fuel, the waste of money and the production of carbon dioxide due to traffic congestion in urban areas are increasing every year.

A great number of methods have been proposed to alleviate traffic congestion. Generally, these solutions can be classified into three major categories:(1) revising the traffic rules to constraint where people could drive, (2) optimizing the existing infrastructure to get a better service, and (3) adjusting traffic signal timing based on different kinds of traffic information. Since the first two categories are relatively law-oriented and time-consuming, this paper focuses on solutions in the third category.

Traffic signal timing optimization is known to be an NP-complete problem [13]. In [2], it has been demonstrated that the traffic signal timing have a strong influence on traffic flow [3]. In this paper, we aim at finding the optimal traffic signal timing that leads to minimum average traffic time for all the vehicles on the roads. In previous works, adjustment based on the proportion of local car flow has been done. However, we show that such local information based adjustment strategy only leads to suboptimal solution on large traffic network. We use GA as our global optimization method and show that it outperforms the local optimization method. According to our experiment results, the solutions obtained by GA reduce the average traffic time by 49% in the case of 1000 vehicles.

However, retrieving detailed information, such as starting point and destination point, from all vehicles is impractical in reality. We, therefore, aim at learning from GA's solution, trying to find the features that are more significant. We use a linear model to approach the global optimization method, showing that this new, local-information based adjustment strategy outperforms the local optimization methods in previous works.

The remainder of this paper is organized as follows. Section II gives an overview of previous works. Section III introduces how we model the problem in our simulation system. Section IV compares the performance between global optimization and local optimization. A new linear traffic model learned from GA is presented in Section V. Section VI concludes this paper.

II. RELATED WORK

Traffic engineering has long been studied. One of the most commonly used frameworks is the Cellular-Automaton (CA). In CA, urban cities are viewed as a composition of regular grid of cells. Through discretizing the space of the cities, we can simplify the behavior of the vehicles and conduct realistic simulation. Several simulation software packages used CA to simulate urban traffic networks, including Simulation of Urban Mobility (SUMO) [7], RoadSim [1] and etc. These simulators have been used to simulate transportation in real cities and help people improve traffic conditions. However, these simulators are too complicated and specific in reaching our goal. To make the simulator simpler, an urban city can be viewed as consisting of only vertical roads, horizontal roads and buildings [16][6][12][5]. Previous work has demonstrated traffic simulation system using such methods. The city model we used is similar to the one proposed by Yasser Hassan [6]. We classified the cells in the grid into three types: buildings, roads and intersection.

Nagel−Schreckenberg model is a classical model for freeway traffic simulation. It defines rules for velocity con-

*These authors contributed equally to this paper trol so that crash between vehicles is avoided. To expand the problem into two-dimension, previous works had proposed several rules on the intersections [8]. The vehicles decide its new direction according to its own destination and the traffic signal when arriving at the intersection. Several works using GA to optimize traffic signal timing were proposed [14][13][15][4]. The results showed that GA does improve traffic condition, especially on large scale traffic network. We therefore use GA as our global optimization method. As for local optimization methods, some works used only local information, such as car flow, the length of waiting queues, to adjust the local traffic signal timing [6]. To minimize the average traffic time at an intersection, the traffic signal timing of each direction should be proportional to the respective car flow.

To this point, several methods for traffic signal timing optimization has been proposed. However, the discrepancy between the performances of the global optimization methods and local optimization methods had not yet been discussed. We aim at verifying that global optimization methods outperform local optimization methods when traffic network becomes larger. Besides, we find new adjustment strategy that can make the result derived by the local optimization method closer to the global optimization result.

III. Problem Modeling

In this section, we present our CA based traffic simulation model. We first define the basic structure of our model. Then, we introduce the rules we employed to simulate all kinds of traffic scenario. At last, we show how vehicles are generated and how we measure their performance.

A. *Cellular Automata Models*

The CA Simulators are based on the CA Theory developed by John Von Neumann [10]. The theory assumes vehicles in the traffic simulation model to be discrete entities. The streets are sampled into a set of points and each point can be occupied by only one vehicle at a time. The *state* of a point is defined as whether a point is occupied by a vehicle or not. For every time slot, all vehicles will move simultaneously according to the traffic scenario and the *state* of each point will be updated. Figure1 shows a simple two-dimensional CA. It consists of regular grid of cells and can be used to simulate a Manhattan-like metropolitan traffic network. The cells in the grid can be classified into three types: Type-B cell, Type-R cell and Type-I cell. The definitions are shown as follow:

- *Type-B* cell represents part of a building, which vehicles cannot run onto.
- *Type-R* cell represents part of a road. Each Type-R cell has a specific direction, which vehicles on this cell can only drive towards.
- *Type-I* cell represents part of an intersection. Vehicles can change their direction only at here. Their decision is based on their destination and the traffic signal condition, which will be elaborated later.

In our experiment, we set the length of the blocks, L_b, to 10. Roads are defined to be two-lane with opposite directions. The length of the borders of the grid, L_g, is calculated by L_b and the number of blocks, N_b, such that

$$L_g = (2 + L_b) \cdot N_b + 2$$

We also define the following parameters for each vehicle:

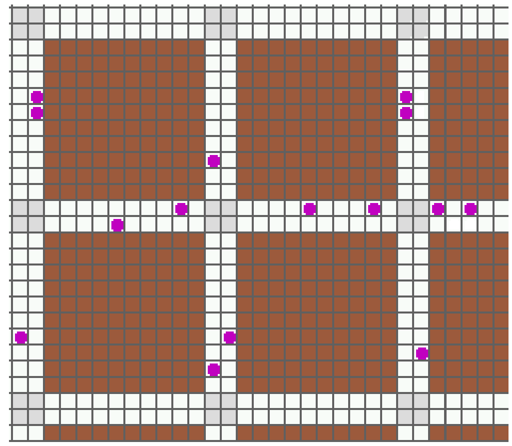

Fig. 1. The structure of simulation model. 10×10 square blocks in brown color are $Type$-B cells, which represent the buildings. 2×2 square blocks in gray color are $Type$-I cells, which represent the intersections. 2×10 rectangles are $Type$-R cells, which represent the roads. The purple dots are the randomly generated cars running on the roads.

- Starting point, $P_s \in [0, L_b]^2$, and destination point, $P_d \in [0, L_b]^2$, are generated from the Gaussian distribution with arbitrary means m_s,$m_d \in [0, L_b]^2$, and variations σ_s,σ_d. For each vehicle, we keep generating a $P_s \sim N(m_s, \sigma_s)$ and a $P_d \sim N(m_d, \sigma_d)$ until these two points are in the range of $[0, L_b]^2$ and are on Type-R cells.
- Current point, $P_c \in [0, L_b]^2$, is initially set to P_s. It will keep changing until the vehicle reaches its P_d.
- Life time, t_f, is the amount of time a vehicle has spent on traveling. It will stop increasing once the vehicle reaches its destination.
- Speed, $v \in [0, 1]$, is the distance a vehicle can move in one time slot.
- Acceleration, $a \in \{0.02, 0.05\}$, is the rate at which the speed of the vehicle changes over each time slot.

Based on the definitions above, we create a simple two-dimensional CA and generate a number of vehicles on it. Through obeying the traffic rules introduced in the next section, our model can simulate most of the traffic scenario in modern city.

B. *Traffic Scenario Simulating*

To efficiently simulate realistic traffic scenario, we define the following rules to simplify the behavior of all vehicles. The rules we applied to all vehicles in our model are presented as follow:

- If a vehicle is on a Type-R cell, the vehicle will keep moving forward unless (1) it reaches its destination point, or (2) the distance between the car in front of it and itself is less than one cell.

- If a vehicle is on a Type-I cell and the corresponding traffic signal is red, the vehicle will turn right or stop according to its destination point.
- If a vehicle is on a Type-I cell and the corresponding traffic signal is green, the vehicle will go straight, turn left, turn right or reverse according to its destination position.

As for the rules defined for speed control are:

- All vehicles have maximum speed of 1. $v \in [0, 1]$.
- A vehicle will keep accelerating if there is no vehicle in front of it or the traffic signal is green. The acceleration is 0.05 point per time slot.
- A vehicle can stop abruptly, that is, its speed can drop to 0 in one time slot. This situation would happen when a vehicle encounters a red traffic signal.
- When a vehicle starts to move from still, its acceleration would be half of usual for four time slots. This penalty is to compensate for the slow start in reality.

After initializing the grid, we generate a number of cars on the grid. According to the traffic scenarios mentioned above, vehicles can start to change its position and speed at each time slot. If a vehicle reaches its destination point, that is, ceil(P_c) = ceil(P_d), it will be eliminated from the grid with its existing time t_f being recorded. An average traffic time can be derived after all the vehicles had reached their destination points. We can use the average time as the score (or the term "Fitness" in GA) of a specific combination of traffic signal cycles.

IV. Optimization Methods

A. Global Optimization (GA)

GA is a well-known optimization method and is shown to have significant improvement on alleviating traffic congestion, especially on large traffic network [4]. In the global optimization scheme, we used GA to optimize the traffic signal timing of all intersections. In this section, we define the chromosome and fitness used in the algorithm and introduce the process in details.

1) Chromosome : We define our chromosome, $\vec{C}_r$, to be a vector which consists of the traffic information of all intersections. N is defined as the number of horizontal roads and vertical roads. For every i, j $\in \{1, 2, \cdots N\}$, there exists an intersection $I_{(i,j)}$ at the crossroad of the i^{th} horizontal road and the j^{th} vertical road. Figure 2 shows the structure of a chromosome corresponding to the map used in our simulation. Each intersection I contains a horizontal green time, G_h, a vertical green time, G_v, a state, S, and a time counter, C. For an intersection $I_{(i,j)}$, $G_{h(i,j)}$ and $G_{v(i,j)} \in \mathbb{Z}$ denote the green time of horizontal and vertical direction at this intersection, respectively. $S_{(i,j)} \in \{0, 1\}$ implies current available direction, with 0 denoting horizontal green, and 1 denoting vertical green. $C_{(i,j)} \in \mathbb{Z}$ is a counter which counts down time for changing the traffic signal state $S_{(i,j)}$. We write $\vec{C}_r$ into:

$$\vec{C}_{r(x)} = \{(G^{(x)}_{v(i,j)}, G^{(x)}_{h(i,j)}, S^{(x)}_{(i,j)}, C^{(x)}_{(i,j)})|i, j \in \{1, 2, \cdots N\}\}.$$

2) Fitness : The fitness is defined as the average traffic time for all vehicles to arrive at their destination points.

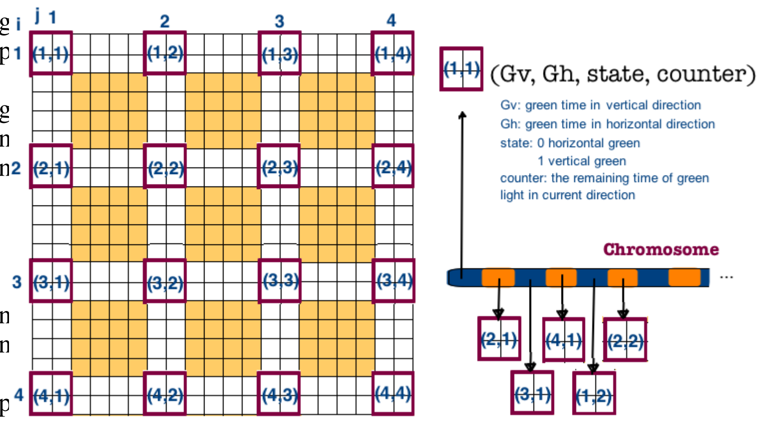

Fig. 2. The structure of our chromosome and the traffic information of every intersection $I_{(i,j)}$, for i, j $\in \{1, 2, \cdots N\}$.

3) Selection : We use the tournament selection without replacement in this step. The number of chromosomes in the mating pool is set to 100. We randomly select 2 chromosomes and put the one with the higher fitness to the mating pool. After this selection process, we will have half number of the total chromosomes, which is 50, in the new chromosome pool. The selection process is done repeatedly for twice so that there will be 100 chromosomes in the new chromosome pool after the whole selection process.

4) Crossover : Two chromosomes, $\vec{C}_{r_{p_1}}$, $\vec{C}_{r_{p_2}}$, are randomly drawn from the chromosome pool as parent chromosomes. Two offspring chromosomes, $\vec{C}_{r_{off_1}}$, $\vec{C}_{r_{off_2}}$, are derived from the crossover process. The crossover process is based on the following equation:

$$\begin{aligned}\vec{C}_{r_{off_1}} &= \alpha \cdot \vec{C}_{r_{p_1}} + (1-\alpha) \cdot \vec{C}_{r_{p_2}} \\ \vec{C}_{r_{off_2}} &= (1-\alpha) \cdot \vec{C}_{r_{p_1}} + \alpha \cdot \vec{C}_{r_{p_2}}.\end{aligned}$$

α is randomly selected from the interval [-0.25,1.25], which is a rule of thumbs. However, this is a simplified version of the crossover process. Detailed operation is shown as follow. Green time of horizontal direction G_h and vertical direction G_v follow the original definition of vector operation. For $i, j \in \{1, 2, \cdots N\}$:

$$G^{(off_1)}_{h(i,j)} = \alpha \cdot G^{(p_1)}_{h(i,j)} + (1-\alpha) \cdot G^{(p_2)}_{h(i,j)}$$

$$G^{(off_2)}_{h(i,j)} = (1-\alpha) \cdot G^{(p_1)}_{h(i,j)} + \alpha \cdot G^{(p_2)}_{h(i,j)}$$

$$G^{(off_1)}_{v(i,j)} = \alpha \cdot G^{(p_1)}_{v(i,j)} + (1-\alpha) \cdot G^{(p_2)}_{v(i,j)}$$

$$G^{(off_2)}_{v(i,j)} = (1-\alpha) \cdot G^{(p_1)}_{v(i,j)} + \alpha \cdot G^{(p_2)}_{v(i,j)}$$

For the initial state S and counter C, the process of crossover is defined as :

$$S^{(off_1)}_{(i,j)} = ceil(\alpha \cdot S^{(p_1)}_{(i,j)}) + (1-\alpha) \cdot S^{(p_2)}_{(i,j)}) \mod 2$$

$$S^{(off_2)}_{(i,j)} = ceil((1-\alpha) \cdot S^{(p_1)}_{(i,j)}) + \alpha \cdot S^{(p_2)}_{(i,j)}) \mod 2$$

$$C^{(off_1)}_{(i,j)} = C^{(off_2)}_{(i,j)} = C^{(p_2)}_{(i,j)}$$

5) Restricted tournament replacement [11]: After the crossover process, each offspring chromosome competes with the parent chromosome with the highest similarity. The similarity of two chromosome $\vec{C}_{r(1)}$,$\vec{C}_{r(2)}$ is defined as follow:
similarity($\vec{C}_{r(1)}$,$\vec{C}_{r(2)}$) =

$$\sum_{i,j=1}^{N}(G^{(1)}_{h(i,j)} - G^{(2)}_{h(i,j)})^2 + (G^{(1)}_{v(i,j)} - G^{(2)}_{v(i,j)})^2 + \lambda \cdot |C^{(1)}_{(i,j)} - C^{(2)}_{(i,j)}|,$$

where λ is a constant which is arbitrarily set to 10 in our experiment.

6) Termination condition: The whole process terminates when chromosomes stop evolving for consecutive 10 iterations.

B. Local Optimization(EM)

In this scheme, we adjust the traffic signal timing with the information from a single intersection. One of the most commonly used information is the flow of cars. In [3], the green time of vertical and horizontal direction should be proportional to the car flow in the corresponding direction in order to reach a minimum average waiting time. We use expectation-maximization (EM) as our optimization method. For a set of traffic signal timing, we obtain the local information of car flow through the simulation. In order to make our simulation more practical in reality, we set a low limitation , $LowLimit$, and a up limitation, $UpLimit$, for every green time. For $i, j \in 1, 2, \cdots N$,

$$G_{v(i,j)}, G_{h(i,j)} \in [LowLimit, UpLimit].$$

1) Decision of green time cycle summation T : For each intersection $I_{(i,j)}$, we define the green time summation T:

$$T_{(i,j)} = G_{v(i,j)} + G_{h(i,j)}.$$

This total time T is determined by the car flow at each intersection. In order to get the car flow information, we randomly generate a combination of traffic signal timing at each intersection and a number of vehicles at the beginning of the test. Two counters, F_v and F_h, are set at each intersection to record the vertical and horizontal car flow. Whenever a vehicle passes through an intersection, we will add a record to the counter in the corresponding direction. After the simulation, we have T be proportional to the car flow summation$(F_v + F_h)$, and scale them into the region of $[2 * lowLimit, 2 * UpLimit]$.

2) Finding the best answers : After deciding the fixed T at each intersection, we simply adjust the traffic light timing on each intersection by the car flow proportion. Since updating traffic signal timing for too many times is not practical in reality, this process is done iteratively for 5 times and the one with the lowest fitness is chosen as the EM method's optimal solution.

C. Comparison between the two methods

For every simulation, we randomly generate a fixed number of vehicles on the grid. Applying GA and EM on the same set of the vehicles, we obtain $Fitness_{GA}$ and $Fitness_{EM}$, the average arriving time of all vehicles for respective methods. We run the simulation for 1000 times and compare the performance of GA to that of EM. The comparison between the performance of these two methods is based on R_w. R_w is defined as follows:

$$R_w = \frac{1}{1000}\sum_{i=1}^{1000}\frac{(Fitness^{(i)}_{EM} - Fitness^{(i)}_{GA})}{Fitness^{(i)}_{GA}}.$$

The superscript (i) denotes the fitness of EM and GA

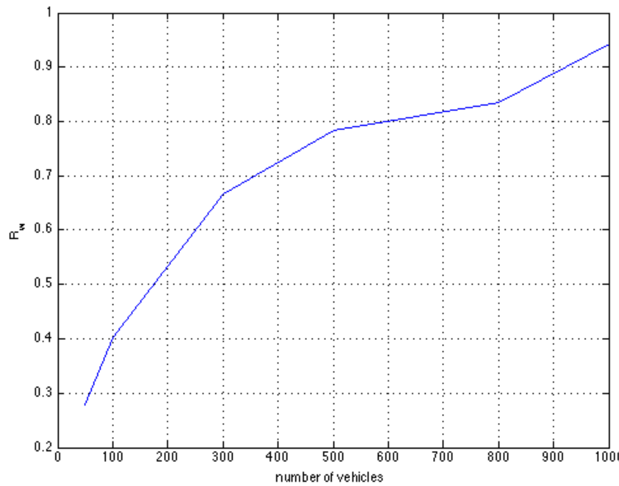

Fig. 3. The difference rate between GA and EM methods is more significant as number of vehicles increases.

TABLE I. R_w TO DIFFERENT NUMBER OF VEHICLES

Car Num	50	100	300	500	800	1000
R_w	0.2779	0.4026	0.6661	0.7842	0.8355	0.9432

derived in the i^{th} simulation. Figure 3 and TableI show the comparison results under different conditions, that is, under different amount of vehicles. The results imply two significant conclusions. First, GA outperforms EM. Since R_w is positive, the average arrival time for vehicles under local optimization, $Fitness_{EM}$, is always longer than the average arrival time for vehicles under global optimization, $Fitness_{GA}$. Hence, to reduce the average arrival time, we must make use of the information other than local car flow. Second, the difference in performance between GA and EM is more significant as the number of vehicles increases. In practical case, applying GA solution on urban areas with high vehicle density will have significant improvement. However, in reality, it is hard to get the detailed information of every starting and destination point for every vehicle. In order to improve the performance of local optimization method, we aim to find the key features and information hidden in GA's solution.

V. Learning From GA

According to our experiment results, the solution obtained by GA has better performance than that derived by local optimization method where recent setting rules of traffic signal timing are based on. However, global information, such as starting and destination points for every single vehicle, is hard to get in practice. Besides, time and memory consumption in running evolutional methods are expensive. In considering these drawbacks in solving practical problem by GA, we apply learning algorithms to the solution obtained by EM based on the solution derived by GA. In this way, we get useful properties which are applied on the original solution derived by EM so that the solution becomes more similar to that of GA. In the following experiment, we prove that the tuned solution of EM performs better than the original solution of EM. In this section, we introduce our method in tuning the sub-optimal solution derived from EM to get a solution with higher performance and show our experiment results.

A. improvement measure method

In our experiments, we conduct t-tests to quantize the level of improvement. A t-test compares whether two groups have different average values. To compare the performance of the solutions derived by EM with the new solutions tuned by learning algorithms, we run both methods on a randomly generated set of vehicles with fixed starting and destination points. The fitness of the solutions derived by both methods are recorded as a pair of results. Each record shown in the following experiment results is derived by calculating t-tests value and transforming the value into p-value on totally 1000 pairs of results. P-value is defined as the probability that the tuned EM method is worse than the original EM method. If p-value $< 5\%$, we say that the new method outperforms EM. Besides, lower p-value shows that the difference between the performances of the two methods is more significant.

B. feature selection

Learning algorithm with linear models are applied on solution derived by EM based on GA's solution. Here we define the traffic information of an intersection as a combination of the green and red signal timings at this intersection, flow of car in and out of the four directions and summation of traffic signal timings at this intersection. Features for an intersection, $I_{(i.j)}$, are extracted from the traffic information of this intersection and its neighborhood intersection, $I_{(i-1.j)}$, $I_{(i+1.j)}$, $I_{(i.j-1)}$ and $I_{(i.j+1)}$. For an intersection $I_{(i.j)}$, useful features are listed as follows:

- $T_{(i,j)}$
- $T_{(i-1,j)}, T_{(i+1,j)}, T_{(i,j-1)}, T_{(i,j+1)}$
- $F_{v(i,j)}, F_{v(i,j)}$
- $F_{h(i-1,j)}, F_{h(i+1,j)}, F_{v(i,j-1)}, F_{v(i,j+1)}$
- $G_{v(i,j)}, G_{h(i,j)}$
- $G_{v(i-1,j)}, G_{v(i+1,j)}, G_{v(i,j-1)}, G_{v(i,j+1)}$
- $G_{h(i-1,j)}, G_{h(i+1,j)}, G_{h(i,j-1)}, G_{h(i,j+1)}$
- $\frac{1}{N^2} \sum_{i,j=1}^{N} G_{v(i,j)}$,

with $T_{(i,j)}$ representing the summation of traffic signal timings of intersection $I_{(i.j)}$ and $F_{v(i,j)}$ representing the car flow on the vertical direction. Our learning target, defined as $Y_{(i,j)}$, is consisting of the vertical green time $G_{v(i,j)}$ and horizontal green time $G_{h(i,j)}$. Ground truth value of $Y_{(i,j)}$ is set to be the solution derived by GA at the position of intersection $I_{(i,j)}$. Feature vector, $X_{(i,j)} \in \mathbb{R}^d$, consists of features we extract from the traffic information of intersection $I_{(i,j)}$ in EM's solution. In this problem, we solve the minimization problem:

$$\underset{w}{\text{minimize}} \quad \sqrt{\sum_{i,j=1}^{N} (Y_{(i,j)} - \langle X_{(i,j)}, w^T \rangle)^2},$$

where $w \in \mathbb{R}^d$ is the corresponding weight vector for feature vector $X_{(i,j)}$. The level of importance of a feature is defined as the absolute value of w_i^*. Less important features are eliminated in order to get a simpler model; important features help us figure out the properties of GA solution.

Every feature is used in training our model in the beginning of the feature selection procedure. Those features with extremely small weight on the corresponding $|w_i *|$ are eliminated. After selecting from the first-stage features, we put the selected features once again into the minimization problem and derive a new solution $w*$ for the modification. To examine the performance of this new solution, 1000 pairs of results are collected through the simulation for the measuring the progress level, which is calculated with $t-$tests method. Secondly, we arbitrarily remove some of the features to see if the improvement level of the new solution degrades or increases. If removing a feature leads to better results in the level of improvement, we will eliminate this feature from our training procedure.

According to our observation in the feature selection process, the most important features extracted from EM's solution in deciding $G^*_{v(i,j)}$ are :

- $G_{v(i-1,j)}, G_{v(i+1,j)}$
- $G_{v(i,j)}$
- $\frac{1}{N^2} \sum_{i,j=1}^{N} G_{v(i,j)}$

For $G^*_{h(i,j)}$, the important features are:

- $G_{h(i,j-1)}, G_{h(i,j+1)}$
- $G_{h(i,j)}$
- $\frac{1}{N^2} \sum_{i,j=1}^{N} G_{h(i,j)}$.

To examine the difference between the solutions derived by GA and EM, we plot the derived $G_{v(i,j)}$ and $G_{v(i,j)}$ of all intersection in a grayscale image. Figure 4 shows the grayscale plot of the traffic signal timing derived by the two methods. The pixels with whiter color in the images represent the part with larger traffic signal timing. The first two images are derived from the vertical and horizontal green time of the solution from EM. In our experiment for EM, traffic signal timing is proportional to car flow in the EM method, which means that these whiter pixels represent the region where rate of car flow is high. From these images, we see that the center part of city has the highest rate of car flow. However, according to Figure 4 (3) (4), the traffic signal timing derived by GA has smaller variance comparing to EM solution.

C. Result of different strategies

In local optimization method, the intersections with high rate of car flow are assigned with a longer green time cycle;

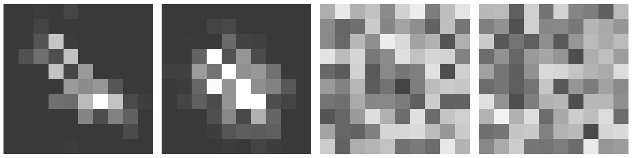

Fig. 4. The grayscale figure of the solution derived by EM and GA. Number of vehicles = 500, map size = 10×10 intersections. From the left to the right are: (1) EM horizontal green time solution (2) EM vertical green time solution (3) GA horizontal green time solution (4) GA vertical green time solution.

the intersections with smaller rate of car flow, usually on the outskirt, are assigned with a shorter green time cycle such that the changing between red and green light is more rapidly. Because there are very few cars running on the outskirt, the influence of outskirt traffic signal timing is less importance than the traffic signal timing in the middle of the city. Considering this, we apply the adjustment only on the intersections that don't have zero rate of car flow on both direction in EM's solution.

After feature selection and training procedure, the best linear function we get is:

$$G^*_{v(i,j)} = 108 - 0.02 \cdot G_{v(i,j)} + 0.02 \cdot G_{v(i,j-1)}$$

$$G^*_{h(i,j)} = 115 - 0.1 \cdot G_{h(i,j)} - 0.01 \cdot G_{h(i-1,j)}$$

for intersection $I_{(i,j)}$ where

$$(G_{v(i,j)} \neq lowLimit) \wedge (G_{h(i,j)} \neq lowLimit) = True.$$

We call this method our Strategy 1. Because the starting points of randomly generated vehicles are mostly closer to the left-top of the map and the destination points are aggregated at the right-bottom, it is reasonable to assume that vertical green timing $G^*_{v(i,j)}$ will have strong relationship to the top neighbor intersection's vertical green time, $G_{v(i,j-1)}$. The linear model we derived also supports this assumption. Car flow of the vertical direction is mainly from the top neighbor intersection. On the other hand, $G_{h(i,j)}$ will have stronger relationship with the left neighborś horizontal green time, $G_{h(i-1,j)}$. To make this more general in some sense, we make a revise version of our Strategy1:

$$G^*_{v(i,j)} = 108 - 0.1 \cdot G_{v(i,j)} - 0.1 \cdot G_{v(i,j-1)}$$

$$G^*_{h(i,j)} = 115 - 0.1 \cdot G_{h(i,j)} - 0.1 \cdot G_{h(i-1,j)}$$

We name this Strategy2. Strategy2 is more symmetric and general in parameter selection. However, the constant term is too specific, causing the model to overfit to our original grid. For even more general and simplicity on the parameter, we further create a smooth version of Strategy2:

$$G^*_{v(i,j)} = 110 - 0.1 \cdot G_{v(i,j)} - 0.1 \cdot G_{v(i,j-1)}$$

$$G^*_{h(i,j)} = 110 - 0.1 \cdot G_{h(i,j)} - 0.1 \cdot G_{h(i-1,j)}$$

The results of using the 3 strategies are shown in Table II

We find that Strategy3, the smoothing version of the method, performs even better than the original one. We further test on the results with different car number on Strategy2 and Strategy3, see Table III and Figure 5. We can see that in the case of using Strategy3, performance increases

TABLE II. P-VALUE ON THE COMPARISON OF EM TO NEW STRATEGIES

Strategy	S1	S2	S3
t-test	-39.2453	-49.701	-67.9299
p-value(%)	0.8109	0.6404	0.4686

as the number of vehicles increases while Strategy2 has even worse performance. Besides, Strategy3 always outperforms Strategy2.

TABLE III. TTEST AND P-VALUE ON STRATEGY 2& STRATEGY3 WITH DIFFERENT NUMBER OF VEHICLES

car num	100	300	500	800	1000
Strategy 2	-43.2537	-49.701	-41.8823	-37.306	-36.3075
	0.7358	0.6404	0.7599	0.8530	0.8765
Strategy 3	-51.9668	-67.9299	-56.9804	-61.1046	-62.9661
	0.6125	0.4685	0.5586	0.5209	0.5055

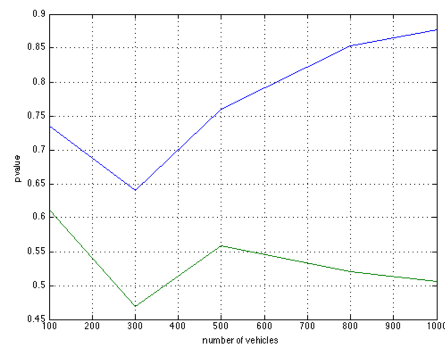

Fig. 5. The p-value of new method comparing to EM.

Smaller p-value indicates that it is less possible for EM to outperform the new methods. In other word, the lower the p-value is, the better the performance of the new method is. According to the result shown in Table II and III, we can see that Strategy3, which is the smooth version of the modification, performs even better than Strategy2, which relies more on the specific parameter derived from EM to GA training. To see how many percent of the shorten time in average arriving time does the new method improve, we calculate the average shortening percentage for the new method versus EM. The result is shown in TableIV. The improvement rate in both strategies have the highest value when car number is set to 500. Although the improvement rate is not an increasing function to the number of cars, the time difference between the solution using these strategies and the solution derived by GA is more significant.

TABLE IV. AVERAGE ARRIVING TIME SHORTENING PERCENTAGE(%) FOR DIFFERENT NUMBER OF VEHICLES AND STRATEGY

car num	100	300	500	800	1000
Strategy 2	4.9960	4.3140	5.4020	3.2680	3.1350
Strategy 3	5.7339	5.8440	7.5100	5.5020	5.5020

D. Generate starting and destination points from multi-mean Gaussian

To examine that our method is general enough in different cases, we further modify the generation of starting and destination point, P_s, P_d, for the vehicles to multi-mean Gaussian distribution. Previously, we choose an arbitrary mean position, m_s, m_d, for the starting and the destination points for a vehicle with variations $sigma_s, sigma_d$. Here we choose two mean positions for both P_s, P_d: $\{m_{s_1}, m_{s_2}\}$ and $\{m_{d_1}, m_{d_2}\}$, respectively. Such distribution of vehicles' starting and destination points leads to a sparser and complicated case of vehicle distribution comparing to previous assumption.

$$P_s \sim N(rand(m_{s_1}, m_{s_2}), \sigma_s)$$
$$P_d \sim N(rand(m_{d_1}, m_{d_2}), \sigma_d)$$

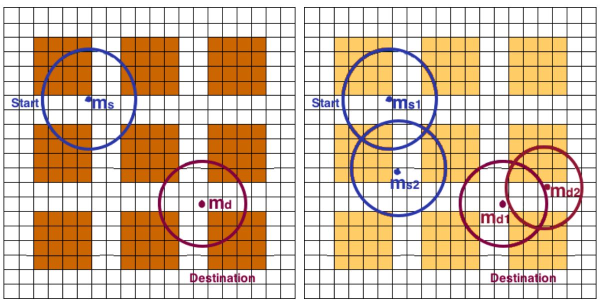

Fig. 6. Plot on the left-hand side represents single mean Gaussian for generating starting and destination points. Right-hand side plot demonstrates multi-mean Gaussian for generating starting and destination points.

We repeat the experiments mentioned in the previous section. Results are shown in Table V and Figure 7.

TABLE V. T-TEST AND P-VALUE ON STRATEGY 2& STRATEGY3 WITH DIFFERENT NUMBER OF VEHICLES

car num	100	300	500	800	1000
Strategy 2	-16.3843	-37.736	-34.6233	-29.6656	-25.1008
	1.9404	0.8433	0.9191	1.0726	1.2675
Strategy 3	-50.8786	-67.7865	-54.9803	-64.6808	-55.0282
	0.6255	0.4695	0.5789	0.4921	0.5784

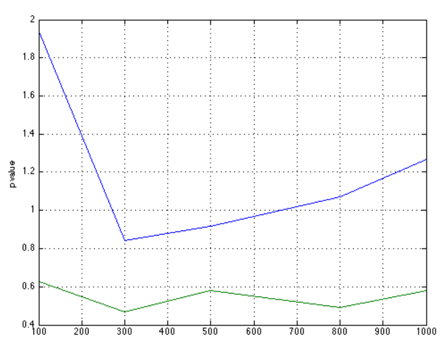

Fig. 7. The p-value of new method comparing to EM.

Clearly we can see that Strategy 3 outperforms Strategy 2. Besides, both methods are within the limitation of p-value $< 5\%$, which means that new methods is better than current traffic signal timing method with high confidence.

TABLE VI. AVERAGE ARRIVING TIME SHORTENING PERCENTAGE(%) FOR DIFFERENT NUMBER OF VEHICLES AND STRATEGY

car num	100	300	500	800	1000
S Strategy 2	1.8310	3.1890	3.0590	2.7750	2.3970
Strategy 3	5.9200	5.9060	7.354	5.7090	5.4890

E. Practical Problem

In practical problem, to find the optimal traffic signal timing in a big city, the whole map is separated into urban and outskirt region. Due to the fact that the traffic signal timing in the urban region has more significant influence on the total traffic timing, only the traffic signal timing in the urban part are adjusted. Firstly, we find the average value of traffic signal timing as the constant term in our method. Then we make small adjustment on the intersection according to the car flow passing through it and its adjacent intersection. For vertical and horizontal traffic signal, we choose the direction from which the vehicles mainly come, and then lower down the time by a proportion. Therefore, for each intersection with vertical green time V and horizontal green time H, we can do the modification as follows:

$$V^* = C - \eta_1 \cdot V - \eta_2 \cdot V_{max(up,down)}$$

$$H^* = C - \eta_1 \cdot H - \eta_2 \cdot H_{max(right,left)}$$

C = average traffic signal timing in a region,

where $V*$ is the new vertical green time, H* is the new horizontal green time. $V_{max(up,down)}$ is the original vertical green time of the up or down neighbor intersection with larger rate of car flow, and $H_{max(right,lrft)}$ is the original horizontal green time of the right or left neighbor intersection with larger rate of car flow. According to our experiment results, we find that improvement in reducing total traffic time is more significant when the number of cars increases. The adjustment strategy is perfect for alleviating traffic jam in big cities during rush hours.

VI. CONCLUSIONS

In this paper, we first demonstrate that the solution derived by GA with global information outperforms the solution obtained by EM with local information. However, it is impractical to retrieve such global information, like vehicles starting positions and destinations, in reality. Besides, building a simulator for a specific city is difficult and time-consuming. Therefore, we try to fine tune the solution obtained by EM method. We propose a linear learning model to tune the solution derived by local optimization method based on the solution obtained by GA. Significant features in our learning model are the information of adjacent intersections including vertical green time, horizontal green time and car flow. Our method has shown to deliver a higher-quality solution than the original EM.

To apply our method to the practical traffic signal timing problem, we first obtain the average traffic signal timing of all intersections. Then, for every intersection, the traffic signal timing is adjusted according to the average traffic signal timing, the flow of cars passing through each intersection and the information of its adjacent intersections.

The adjustment strategy is as follows. First, in the region where the value of the flow of cars are higher, more vehicles will be affected by the red traffic light on an intersection. Thus, traffic signal timing should be lowered to reduce the waiting time at each intersection and therefore alleviate congestion. Second, shortening traffic signal timing on the outskirt is a way to alleviate traffic congestion in the urban area, since it will increase the probability for vehicles to pass through the outskirt rather than the center of the city. We can take this as a way to encourage drivers taking the advantages of the outskirt ways and avoid getting into the crowded city center.

REFERENCES

[1] M. M. Artimy, W. Robertson, and W. J. Phillips. Connectivity in inter-vehicle ad hoc networks. *IEEE Canadian conf.Elec Comp.Eng.*, 1:293–298, 2004.

[2] E. Brockfeld, R. Barlovic, A. Schadschneider, and M. Schreckenberg. Optimizing traffic lights in a cellular automaton model for city traffic. *Phys Rev E Stat Nonlin Soft Matter Phys*, 64(5 Pt2)(056132), 2001.

[3] J. C. Burguillo-Rial, P. S. Rodríguez-Hernández, E. Costa-Montenegro, and F. Gil-Castiñeira. Optimizing traffic lights in a cellular automaton model for city traffic. *Computing and Informatics*, 28:1001–1012, 2009.

[4] H. Ceylan and M. G .H. Bell. Traffic signal timing optimization based on genetic algorithm approach, including drivers' routing. *Transportation Research Part B*, (38):329–342, 2004.

[5] J. Esser and M. Schreckenberg. Microscopic simulation of urban traffic based on cellular automata. *International Journal of Modern Physics*, 8(5):1025–1036, 1997.

[6] Y. Hassan and E. Tazaki. Emergence decision using hybrid rough sets/cellular automata. *Kybernetes*, 35(6):797–813, 2006.

[7] D. Krajzewicz, G. Hertkorn, P. Wagne, and C. Rössel. Sumo (simulation of urban mobility): An open-source traffic simulation. *4th Middle East Symp. Simulation Modeling*, pages 183–187, 2002.

[8] N. Linesch. A traffic analysis using a two dimensional cellular automata model. *PHY199*, 2006.

[9] T. Lomax, D. Schrank, and B. Eisele. 2012 annual ur- ban mobility report. *Texas A&M Transportation Institute URL: http://mobility.tamu.edu/ums/*, 2012.

[10] J. V. Neumann. Probabilistic logic and the synthesis of reliable organisms from unreliable components. *Automata Studies, Princeton University Press*, 1956.

[11] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. *Genetic and Evolutionary Computation Conference (GECCO)*, pages 511–518, 2001.

[12] B. Saduon. Efficient simulation methodology for the design of traffic lights at intersections in urban areas. *Simulation*, 79:243–251, 2003.

[13] J. J. Sanchez, M. J. Galan, and E. Rubio. Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization. *The Congress on Evolutionary Computation(CEC)*, 2:1668–1674, 2004.

[14] Z. Shen and K. Wang. Agent-based traffic simulation and traffic signal timing optimization with gpu. *Intelligent Transportation Systems (ITSC)*, pages 145–150, 2011.

[15] L. Singh, S. Tripathi, and H. Arora. Time optimization for traffic signal control using genetic algorithm. *International Journal of Recent Trends in Engineering*, 2(2), 2009.

[16] O. K. Tonguz, W. Viriyasitavat, and F. Bai. Modeling urban traffic: a cellular automata approach. *IEEE Communications Magazine*, 47:142–150, 2009.