Real-Time Traffic Signal Control for Roundabouts by Using a PSO-Based Fuzzy Controller

Yue-jiao Gong, Jun Zhang (Corresponding Author)
Dept. of C.S., Sun Yat-sen University
Key Laboratory of Digital Life, Ministry of Education
Key Laboratory of Software Technology, Education Dept. of Guangdong Province, P.R. China
junzhang@ieee.org

***Abstract*—Developing traffic signal control methods is considered as the most important way to improve the traffic efficiency of modern roundabouts. This paper applies a traffic signal controller with two fuzzy layers for signalizing roundabouts. The outer layer of the controller computes urgency degrees of all the phase subsets and then activates the most urgent subset. This mechanism helps to instantly respond to the current traffic condition of the roundabout so as to improve real-timeness. The inner layer computes extension time of the current phase and decides whether to turn to the next phase in the running phase subset. As the phase sequences are well-designed, the inner layer smoothes the traffic flows which helps to avoid traffic jam. An offline particle swarm optimization (PSO) algorithm is developed to optimize the membership functions adopted in the proposed controller. In this way, the membership functions in the controller are no longer given by human experience, but provided by the intelligent algorithm. Simulation results demonstrate that the proposed controller outperforms previous traffic signal controllers in terms of improving traffic efficiency of modern roundabouts.**

Keywords-fuzzy logic, membership function, particle swarm optimization (PSO), roundabout, traffic control.

I. INTRODUCTION

With the rapid development of automobile industry and the increase in urban population, traffic congestion is becoming a critical problem in large urban cities all around the world. As existing traffic facilities can hardly be extended due to the cost and environmental issues, traffic engineers are paying more attention on developing traffic signal control methods to better use the available facilities so as to provide better service and prevent congestion.

With the spread of inexpensive sensors and communication devices, real-time traffic data can be easily accessed. In recent years, researchers have made great efforts to develop real-time traffic signal controllers [1]-[10]. The vehicle actuated method [1] was the first real-time traffic controller. It decided whether to extend the current phase according to whether there were vehicles detected in real time. The method was simple, and it was effective when the traffic condition was not heavy. For further study, real-time traffic control based on fuzzy logic has become a very hot research field [2]-[8], because it has been known that fuzzy logic is well suited to control complex systems with uncertainties and human perception. Basically, according to the control mechanisms, there were two types of fuzzy traffic signal controllers in use. The first type outputs an extension time of the current green phase [2]-[4]. If the extension time was bigger than a predefined threshold value, the current green phase would last another period of time. Otherwise, it would turn to the next phase in the phase circle. We term this fuzzy control method as FUZZY-TURN. The other type output the urgency degrees of all the phases at set intervals [5]-[7]. The phase would jump to the phase which was in the most urgent by the end of each interval. This fuzzy control method can be termed FUZZY-JUMP. The work in [8] is a developed FUZZY-JUMP controller. The controller activates the most urgent phase, but applies an additional stage to calculate the last time for the activated phase.

As the FUZZY-TURN controller uses well-designed phase sequences that only permit consistent adjacent phases, it is not easy to cause traffic jams. But the use of preset phase sequences may lower the real-timeness of the controller. On the other side, FUZZY-JUMP is capable of frequently changing its phase sequences and therefore has better real-timeness than FUZZY-TURN. However, the FUZZY-JUMP controller enables inconsistent traffic flows of adjacent phases exist simultaneously and is more likely to cause traffic jams. Moreover, frequently changing sequence of phases may confuse the drivers and lead to traffic accidents.

In our preliminary work [11], a mixed fuzzy traffic control mechanism termed FUZZY-MIX is proposed. In the FUZZY-MIX controller, the phase set is divided into several phase subsets according to the directions. The FUZZY-MIX has two layers of fuzzy controllers. The outer layer computes the urgency degrees of different phase subsets and determines the next running subset, while the inner layer computes the extension time of the green phase and determines whether to turn to the next phase in the running phase subset. By this way, the FUZZY-MIX technique is a combination of FUZZY-JUMP and FUZZY-TURN methods. The FUZZY-MIX controller incorporates the advantages of FUZZY-TURN and FUZZY-JUMP with the outer layer to improve real-timeness and the inner layer to reduce the risk of traffic jam and improve traffic safety.

But in a fuzzy controller, the traditional use of membership functions given by human preference cannot guarantee the optimal performance of the controller [2][7][8]. The

This work was supported in part by the The National Science Fund for Distinguished Young Scholars No.61125205, National Natural Science Foundation of China No.61070004 and NSFC Joint Fund with Guangdong under Key Project U0835002.

evolutionary computation (EC) and swarm intelligence (SI) technique is developing very fast recently [12]-[14]. In order to further improve the performance of FUZZY-MIX, this paper develops particle swarm optimization (PSO) to optimize the membership functions adopted in the controller. The PSO algorithm was firstly introduced by Kennedy and Eberhart in 1995 [15][16]. The algorithm simulates the swarm behaviors of bird flocking or fish schooling and is very easy to implement. In recent years the algorithm has been successfully applied on various problems and demonstrated to have high operating efficiency [17]-[20]. PSO is suitable for solving optimization problems, especially continuous optimization problems. In this paper, the PSO algorithm is applied to optimize the shapes of the 11 membership functions used in FUZZY-MIX (six for input variables and five for output variables). In this way, a novel PSO-based optimal fuzzy controller for modern roundabouts termed FUZZY-MIX-OPT is developed. In the proposed FUZZY-MIX-OPT controller, the shapes of membership functions are no longer given by human preference, but optimized by the intelligent algorithm inartificially.

In the experiment, a common roundabout with two circulatory lanes and four approaches is simulated, and 16 traffic conditions including eight steady conditions and eight time-varying conditions are tested. We implement five real-time controllers including the vehicle actuated method termed VA [1], the FUZZY-TURN method [2]-[4], the FUZZY-JUMP method [5]-[7], the FUZZY-MIX controller [11], and the proposed FUZZY-MIX-OPT controller on the roundabout. Simulation results and comparisons show that the FUZZY-MIX controller outperforms the previous traffic controllers VA, FUZZY-TURN, and FUZZY-JUMP. In addition, by using the PSO algorithm to optimize the membership functions of FUZZY-MIX, the PSO-based FUZZY-MIX-OPT controller further improves the performance of FUZZY-MIX and provides a very effective and efficient traffic signal control tool for the roundabout.

The rest of this paper is organized as follows. Section II introduces the geometric design and phase composing of the roundabout. Section III describes the methods of fuzzy traffic controllers. In Section IV, the PSO algorithm is applied on optimizing the membership functions of the fuzzy traffic controller. In Section V, simulations are conducted and comparisons of five traffic controllers for the roundabout are made. At last, a conclusion is drawn in Section VI.

II. ROUNDABOUT MODEL

A. Geometric Design

The geometric design of a commonly employed modern roundabout is illustrated in Fig. 1. The roundabout is composed of four approaches numbered 0, 1, 2, and 3. For each approach, there are three entrance lanes. From left to right, the three entrance lanes only permit the left-turn vehicle flow, the go-through vehicle flow, and the right-turn vehicle flow, respectively. Stop lines are placed before the left-turn and go-through entrance lanes. For a signalized roundabout, the traffic signal controls the movement of vehicles behind the stop lines. The right-turn traffic flows are free from the traffic signal

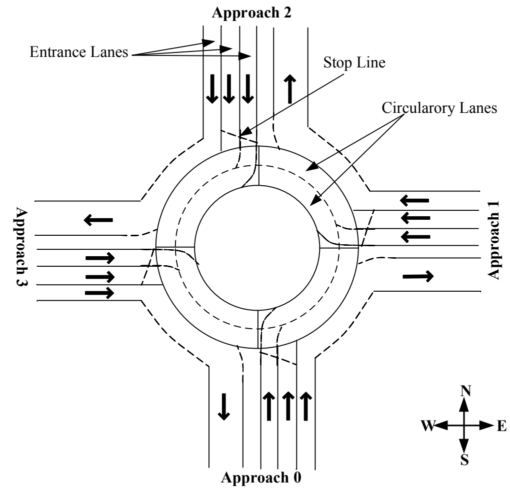

Figure 1. Illustration of the roundabout.

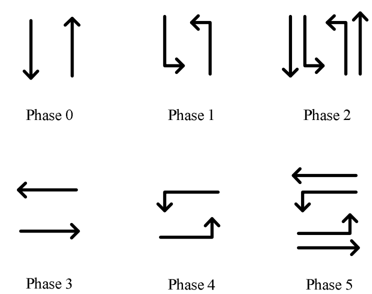

Figure 2. Phase set of an ordinary four-approach roundabout.

because they can turn right directly without entering the circulatory lanes of the roundabout. The roundabout contains two circulatory lanes. The inner one is used for left-turn movements, while the outer one is occupied by go-through vehicles and the left-turn vehicles which are about to enter or left the inner lane.

B. Traffic Signal

As described in Subsection A, there are totally eight traffic flows consist of the left-turn and go-through flows of the four approaches under the control of traffic signal. Six possible phases for the eight traffic flows are illustrated in Fig. 2, which are described as follows.

Phase 0: the go-through traffic flows on approaches 0 and 2 are activated, whereas the other traffic flows are stopped.

Phase 1: only the left-turn traffic flows on approaches 0 and 2 are activated.

Phase 2: all the traffic flows on approaches 0 and 2 get green time.

Phase 3: green time for approaches 0 and 2 ends. The go-through traffic flows on approaches 1 and 3 are activated.

Phase 4: the left-turn flows on approaches 1 and 3 start to move whereas the other flows are stopped.

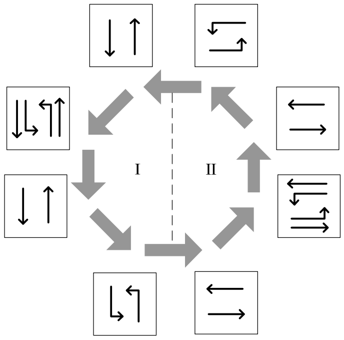

Figure 3. Phase circle of the four-approach roundabout.

Phase 5: green time for all the traffic flows on approaches 1 and 3.

The phase set can be divided into two subsets. Phases 0, 1, and 2 belong to subset I which permit the north-south traffic flows and stop the west-east traffic flows. Phases 3, 4, and 5 do just the opposite and belong to subset II.

As shown in Fig. 3, a phase circle is a cycle of the phase sequences. It is well-designed, only permitting consistent traffic flows in adjacent phase sequences to avoid traffic jam. Traditional traffic signal timing is to determine how long each phase involved in the circle should last. The fixed-time control methods use fixed durations for the phases, whereas the real-time control methods use time-varying durations for the phases. However, using preset phase sequences may lower the real-time response of the system. Thus, instead of using the phase circle, some state-of-the-art real-time signal controllers always jump to the phase which is in the most urgent so as to improve the real-timeness. However, as this mechanism may lead to occurrences of inconsistent traffic flows in adjacent phases (e.g. if the system jumps instantly from phase 0 to phase 5), the system is vulnerable to traffic jam. Employing an all-red time between phase switching so as to clear the vehicles in the roundabout can help to reduce the risk of traffic jam [21][22]. But frequently inserting all-red time induces additional time cost and reduces the traffic capacity of the roundabout during a time period.

III. Fuzzy Traffic Controller

An illustration of using a fuzzy traffic controller (FTC) to signalize a roundabout is shown in Fig. 4. Inputting the current traffic condition, the FTC outputs the selected active phase to the traffic signal module. According to the selected phase, the signal of each approach permits a portion of vehicles behind the stop lines to enter the circulatory lanes. Then the traffic condition of the roundabout changes and the FTC generates the newly selected active phase accordingly.

In our preliminary work [11], we have developed a two-layer FTC hybridizing FUZZY-TURN and FUZZY-JUMP, which is termed FUZZY-MIX. In this section, brief descriptions of the FUZZY-TURN, FUZZY-JUMP, and FUZZY-MIX controllers are presented. All the three controllers receive same input variables including queue length (*QL*) and waiting time (*WT*) which reflect the current traffic condition. The control mechanisms of the three controllers differ with respect to their output variables, which are described as follows.

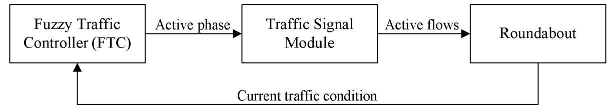

Figure 4. Illustration of using FTC to signalize roundabout.

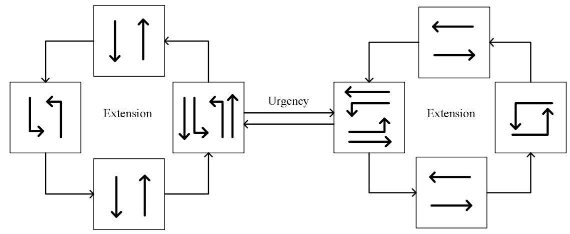

Figure 5. Phase sequences of FUZZY-MIX controller.

FUZZY-TURN: by the end of each phase, the FUZZY-TURN generates an extension time (*ET*) of the current phase according to the traffic condition of the roundabout. If *ET* is larger than a predefined threshold value (Φ time units), the green phase lasts for another *ET* period of time. Otherwise, the current green phase is terminated and the next phase in the phase circle (shown in Fig. 3) is activated and allocated with an initial duration (Θ time units).

FUZZY-JUMP: at set intervals (Δ time units), the urgency degree (*UD*) of each phase is computed. Then the signal module jumps to the phase with largest *UD*. Obviously, if the current green phase has the largest *UD*, it lasts for another Δ period of time; otherwise it is replaced by another phase. What should be pointed out is that if the adjacent phases are inconsistent, an all-red time (θ time units) should be inserted between the two phases to avoid traffic jam.

FUZZY-MIX: the phase set is divided into two subsets as presented in Section II-B. The FUZZY-MIX controller consists of two fuzzy layers. The outer layer is a coarser-grained form of the FUZZY-JUMP controller. By the end of each phase, this layer computes *UD* of the two subsets.

1) If *UD* of the current running subset is larger than that of the other subset, the controller performs the inner layer. The inner layer is similar to the FUZZY-TURN controller, which outputs *ET* of the current phase. If *ET* is larger than Φ, the current phase is maintained. Otherwise, the inner layer activates the next phase in the current subset and allocates it with the initial duration Θ. Fig. 5 shows the phase sequences of the two subsets.

2) If *UD* of the current running subset is smaller than that of the other subset, the entrance phase of the other subset is activated. To avoid inconsistent traffic flows existing in the roundabout, all-red time θ is inserted between the phase conversions. It can be observed in Fig. 5 that the entrance phase of subset I is phase 1 whereas that of subset II the phase 4. The two phases are with maximal number of active flows in

order to maximally relieve the traffic burden of the roundabout after the all-red time.

Detailed implementations of the FUZZY-MIX FTC can be found in [11]. In general, each layer of FUZZY-MIX follows the universal procedure of fuzzy logic controllers, which consists of a fuzzifier, a fuzzy rule base, a fuzzy inference engine, and a defuzzifier. The fuzzifier receives crisp input values and applies membership functions of the input variables to generate linguistic values with membership grades. Then, the fuzzy inference engine receives the linguistic values and infers the linguistic outputs according to related rules in the fuzzy rule base. At last, the defuzzifier is executed to convert linguistic outputs with membership grades into a single crisp output value.

IV. PSO-BASED FUZZY TRAFFIC CONTROLLER

In industrial applications of fuzzy logic controller, the shapes of membership functions are always chosen by human arbitrarily. It is based on engineers' experience, and cannot guarantee to provide optimal control for the corresponding system. In this paper, to further improve the performance of the FUZZY-MIX controller, a PSO algorithm is adopted to optimize the shapes of the membership functions. The FUZZY-MIX controller with optimal membership functions trained by PSO is then termed FUZZY-MIX-OPT. This Section describes the implantations of applying PSO algorithm for optimizing the membership functions of FUZZY-MIX-OPT.

A. Methodology

PSO algorithm stimulates the foraging behavior of birds or fishes, in which a group of particles are randomly scattered in the problem space and search for the optimal point simultaneously. In PSO algorithm, a particle swarm composed of M particles is maintained. Each particle is associated with a position vector $\boldsymbol{X}_i = [x_i^1, x_i^2, ..., x_i^N]$, a velocity vector $\boldsymbol{V}_i = [v_i^1, v_i^2, ..., v_i^N]$, and a fitness value Fit_i (where N is the dimensionality of the search space and $i = 1, 2, ..., M$). The position vector $\boldsymbol{X}_i$ stands for a candidate solution of the optimization problem, and is evaluated to obtain the fitness value of a particle. Each particle maintains a previous best position vector $\boldsymbol{pBest}_i = [pBest_i^1, pBest_i^2, ..., pBest_i^N]$ as its search experience. Meanwhile, the global best position found by the whole swarm during the search process is recorded as $\boldsymbol{gBest} = [gBest^1, gBest^2, ..., gBest^N]$. Then, particle i adjusts its flying velocity based on $\boldsymbol{pBest}_i$ (self-cognitive) and $\boldsymbol{gBest}$ (social-cognitive), and accordingly updates its position vector. In this way, the particles tend to fly towards better and better domain. Eventually, the particle swarm is likely to converge to the optimal position of the problem space. PSO is capable of exploring and exploiting some large, complex, and initially unknown problem spaces, which traditional algorithms (enumerative, heuristic, etc.) can hardly deal with. The algorithm is suitable to optimize the membership functions of the fuzzy traffic controller.

The procedure of the PSO algorithm for optimizing membership functions of FUZZY-MIX-OPT is very simple as shown in Fig. 7. In the initialization, the position and velocity vectors of the M particles are randomly generated. The fitness values of the particles are evaluated according to (1). The $\boldsymbol{pBest}_i$ of particle i is set with its initial position vector $\boldsymbol{X}_i$, and then the $\boldsymbol{gBest}$ is set with the currently best $\boldsymbol{pBest}_i$.

In each iteration of the algorithm, the particles in the swarm interact with each other and collaborate to search the problem space. The update of position and velocity for particle i is defined as

$$\begin{aligned} \boldsymbol{V}_i = \omega \boldsymbol{V}_i + c_1 \boldsymbol{rand}_1 \otimes (\boldsymbol{pBest}_i - \boldsymbol{X}_i) \\ + c_2 \boldsymbol{rand}_2 \otimes (\boldsymbol{gBest} - \boldsymbol{X}_i) \end{aligned}, \tag{2}$$

$$\boldsymbol{X}_i = \boldsymbol{X}_i + \boldsymbol{V}_i, \tag{3}$$

where ω is the inertia weight; c_1 and c_2 are acceleration coefficients which determine the relative weights of self-cognitive and social-influence; $\boldsymbol{rand}_1$ and $\boldsymbol{rand}_2$ are two random vectors uniformly distributed over $[0, 1]^N$; $\otimes$ is component-wise multiplication.

Then, the positions of all the particles are evaluated. If the new position of particle i is better than its previous best position, $\boldsymbol{pBest}_i$ is replaced by $\boldsymbol{X}_i$. Furthermore, if a new global best-so-far position is discovered, the $\boldsymbol{gBest}$ is accordingly updated. After a few iterations, the particle swarm converges, and an optimal or near-optimal solution of the problem is

TABLE I. SIXTEEN TRAFFIC CONDITIONS

Condition		Arrival Rate (vehicle/unit)							
		0-L	1-L	2-L	3-L	0-S	1-S	2-S	3-S
Steady	C1	0.102	0.097	0.092	0.123	0.118	0.108	0.108	0.125
	C2	0.156	0.099	0.143	0.102	0.176	0.111	0.180	0.108
	C3	0.067	0.074	0.068	0.068	0.178	0.149	0.158	0.169
	C4	0.057	0.111	0.069	0.121	0.154	0.199	0.142	0.212
	C5	0.178	0.165	0.189	0.215	0.199	0.212	0.203	0.222
	C6	0.181	0.121	0.243	0.132	0.209	0.131	0.255	0.143
	C7	0.132	0.114	0.117	0.108	0.251	0.232	0.223	0.209
	C8	0.101	0.188	0.098	0.200	0.198	0.287	0.216	0.320
Time-varying	C9	0.089 ↓ 0.178	0.102 ↓ 0.177	0.103 ↓ 0.190	0.077 ↓ 0.168	0.112 ↓ 0.198	0.108 ↓ 0.189	0.131 ↓ 0.179	0.105 ↓ 0.170
	C10	0.057 ↓ 0.224	0.066 ↓ 0.232	0.072 ↓ 0.189	0.072 ↓ 0.250	0.080 ↓ 0.200	0.066 ↓ 0.240	0.073 ↓ 0.232	0.081 ↓ 0.255
	C11	0.156 ↓ 0.223	0.078 ↓ 0.182	0.123 ↓ 0.189	0.050 ↓ 0.132	0.158 ↓ 0.240	0.089 ↓ 0.194	0.149 ↓ 0.278	0.101 ↓ 0.201
	C12	0.101 ↓ 0.242	0.058 ↓ 0.179	0.097 ↓ 0.199	0.034 ↓ 0.155	0.125 ↓ 0.252	0.077 ↓ 0.177	0.102 ↓ 0.280	0.073 ↓ 0.156
	C13	0.101 ↓ 0.158	0.097 ↓ 0.135	0.108 ↓ 0.138	0.099 ↓ 0.160	0.159 ↓ 0.277	0.188 ↓ 0.256	0.180 ↓ 0.280	0.176 ↓ 0.310
	C14	0.057 ↓ 0.158	0.077 ↓ 0.170	0.079 ↓ 0.120	0.100 ↓ 0.155	0.140 ↓ 0.300	0.161 ↓ 0.341	0.151 ↓ 0.400	0.130 ↓ 0.299
	C15	0.055 ↓ 0.101	0.089 ↓ 0.151	0.048 ↓ 0.078	0.100 ↓ 0.162	0.121 ↓ 0.178	0.189 ↓ 0.298	0.148 ↓ 0.210	0.210 ↓ 0.250
	C16	0.055 ↓ 0.158	0.102 ↓ 0.189	0.077 ↓ 0.176	0.134 ↓ 0.188	0.160 ↓ 0.298	0.205 ↓ 0.315	0.144 ↓ 0.298	0.245 ↓ 0.341

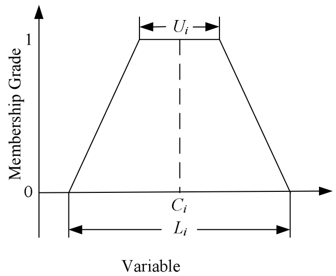

Figure 6. Representation of a trapezoidal function.

obtained. The PSO algorithm is always defined with a maximal number of iterations G, by which the algorithm will be terminated.

B. Representation

In order to use PSO algorithm for optimizing the membership functions in FUZZY-MIX-OPT, we should define the representation of particle's position and the search space. Each particle's position should indicate a candidate solution of the problem, i.e., the shapes of all the membership functions adopted in FUZZY-MIX-OPT.

As described in [11], there are totally 11 membership functions in the FUZZY-MIX controller (three for *QL*, three for *WT*, two for *ET*, and three for *UD*.). All the membership functions are triangular or trapezoidal functions, which are commonly used in fuzzy logic controllers. Moreover, a triangular function can be regarded as a special trapezoidal function (the length of upper line is 0). Therefore, only trapezoidal function is concerned in the optimization.

As shown in Fig. 6, each membership function in FUZZY-MIX-OPT can be represented by a triple $<U_i, L_i, C_i>$, where U_i and L_i stand for the length of upper line and lower line respectively, and C_i is the coordinate of the middle point of the parallel sides. It should be noticed that each membership function is regarded as an isosceles trapezoid in order to reduce the search space of the PSO algorithm. Moreover, in each trapezoidal function there exists a constraint that U_i should be no larger than L_i. We can eliminate this constraint by introducing a variable $D_i = L_i - U_i$ and representing each trapezoidal function as $<U_i, D_i, C_i>$. In the representation of PSO algorithm, each candidate solution (particle's position) is in a form of $[U_1 D_1 C_1 U_2 D_2 C_2 \dots U_{11} D_{11} C_{11}]$, which indicates the shapes of the 11 membership functions in the FTC. Consequently, the dimensionality of the search space of the PSO algorithm is 33. In addition, the ranges of all the variables are presented as follows,

$$0 \le U_i \le 10,\ 0 \le D_i \le 10,\ 0 \le C_i \le 20\ (i = 1,2,3),$$
$$0 \le U_i \le 50,\ 0 \le D_i \le 50,\ 0 \le C_i \le 100\ (i = 4,5,6),$$
$$0 \le U_i \le 7.5,\ 0 \le D_i \le 7.5,\ 0 \le C_i \le 15\ (i = 7,8),$$
$$0 \le U_i \le 0.5,\ 0 \le D_i \le 0.5,\ 0 \le C_i \le 1\ (i = 9,10,11).$$

C. Evaluation

In the literature, the fitness value of each particle is commonly evaluated by a mathematical function (called fitness function) defined according to the objective of the problem. However, it is hard to define a mathematical fitness function for a traffic signal control system. Therefore, in this paper we perform simulations and evaluate each particle's position by the performance measure during the simulation.

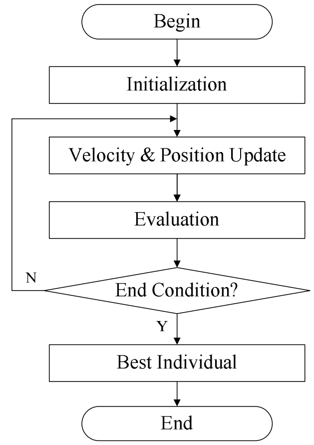

Figure 7. Flowchart of PSO algorithm.

In the algorithm, Fit_i of particle i is determined by

$$Fit_i = w_1 \times (vehMiss / vehPass) + w_2 \times vehDelay \qquad (1)$$

where $vehMiss/vehPass$ is the ratio of the number of undetected vehicles (*vehMiss*) to the number of passing vehicles (*vehPass*) during the simulation; *vehDelay* stands for the average delay time of vehicles; w_1 and w_2 are the corresponding weights for $vehMiss/vehPass$ and *vehDelay*. Details of the simulation will be described in Section V.

V. Simulation Results and Comparisons

In the experiment, the performance of different controllers is compared by simulations conducted on the roundabout with two circulatory lanes and four approaches. In order to comprehensively investigate the performance of these controllers for signalizing the roundabout, we generate 16 traffic conditions for the roundabout, each of which has its own characteristics. Then, five real-time traffic controllers including the vehicle actuated method (VA) [1], the FUZZY-TURN controller, the FUZZY-JUMP controller, the FUZZY-MIX controller, and the FUZZY-MIX-OPT controller are applied. In this Section, we first describe the simulation environment and parameters, and then present the simulation results and comparisons.

A. Simulation Environment and Parameters

We simulate the roundabout described in Section II with Visual C++. The program is running on a machine with Intel Pentium Dual CPU, 1.99 GHz/500 MB of RAM. Table I shows the vehicle arrivals (subject to Poisson distribution [8]) on different lanes of the 16 traffic conditions, where 0-L stands for the left-turn vehicle arrival rate on approach 0, 1-S represents the go-through vehicle arrival rate on approach 1, and so forth. For the time-varying conditions C9-C16, the vehicle arrival rate of each lane starts with an initial value and

TABLE II. NUMBER OF UNDETECTED VEHICLES OBTAINED BY THE FIVE CONTROLLERS

Condition	VA	FUZZY-TURN	FUZZY-JUMP	FUZZY-MIX	FUZZY-MIX-OPT
C1	0	0	0	0	0
C2	0	4	1	0	0
C3	0	0	0	0	0
C4	7	10	1	0	0
C5	7988	2361	825	0	0
C6	1248	1345	751	19	0
C7	1168	239	104	0	0
C8	8343	5392	4603	69	0
C9	161	5	24	0	0
C10	3474	1250	858	8	0
C11	1732	752	460	4	0
C12	1242	509	409	13	0
C13	4242	1082	856	0	0
C14	8908	4053	4038	715	77
C15	404	161	197	4	0
C16	10708	4530	4508	19	3

TABLE III. AVERAGE VEHICLE DELAY OBTAINED BY THE FIVE CONTROLLERS

Condition	VA	FUZZY-TURN	FUZZY-JUMP	FUZZY-MIX	FUZZY-MIX-OPT
C1	19.990	19.944	22.813	16.204	11.477
C2	22.009	21.546	22.619	15.572	12.314
C3	20.881	20.076	20.703	16.371	11.638
C4	22.588	21.239	21.015	16.068	11.911
C5	40.185	36.966	31.220	15.813	13.752
C6	30.952	29.010	27.002	15.546	13.430
C7	30.573	25.506	25.187	15.422	13.242
C8	37.262	35.551	35.225	19.164	14.230
C9	24.413	22.165	23.891	15.719	12.432
C10	30.037	28.255	28.205	16.410	12.973
C11	29.345	27.005	26.234	15.438	13.014
C12	27.457	25.347	25.532	15.703	12.659
C13	32.079	27.376	27.709	15.673	13.270
C14	33.356	29.278	31.894	17.527	14.528
C15	26.098	23.382	23.014	16.276	12.535
C16	36.276	32.705	34.017	16.763	14.347
Mean	**28.969**	**26.584**	**26.642**	**16.229**	**12.985**

gradually increases to the final value. The differences among the 16 traffic conditions are presented as follows.

1. The flow rates of C1-C8 are steady whereas those of C9-C16 are time-varying.

2. In C1, C3, …, C15, the differences between the north-south flow rates (the arrival rates of vehicles from approaches 0 and 2) and the west-east flow rates (the arrival rates of vehicles from approaches 1 and 3) are inconspicuous. In contrast, in C2, C4, …, C16, those differences are conspicuous.

3. In C1, C2, C5, C6, C9, C10, C13, and C14, the left-turn flow rate and go-through flow rate of a same approach are similar. In contrast, the left-turn and go-through flow rates of a same approach differ a lot in the other conditions.

4. Among the steady traffic conditions, C1-C4 are light conditions the vehicle arrival rates of which are relatively low, whereas C5-C8 are heavy conditions which have much higher vehicle arrival rates than C1-C4.

5. Among the time-varying conditions, the range of the flow rate in C13-C16 is larger than that in C9-C12.

It is assumed that the detectors are installed at a certain distance to the roundabout, and the maximal number of detectable queuing vehicles of each entrance lane is equal to 20. If the queue length of an entrance lane exceeds 20, the newly arrived vehicle on this lane will be regarded as an undetected vehicle. We always expect to reduce the length of waiting-vehicle queue and the number of undetected vehicles (*vehMiss*). Moreover, the undetected vehicles lead to occurrence of detecting error which would do harm to the performance of the fuzzy controller. Therefore, when using PSO algorithm to optimize the membership functions of FUZZY-MIX-OPT, minimizing the *vehMiss* should be concerned. Due to the random factors in the simulation, minimizing the ratio of *vehMiss* to *vehPass* is used as the first objective of PSO, which is shown in equation (1). Moreover, as a common performance index of traffic signal control, the average delay time of vehicles is used as the second objective of PSO. In the experiment, we set w_1 and w_2 of equation (1) as $w_1=1$ and $w_2=10^{-8}$ so as to make minimizing the *vehMiss*/*vehPass* in preference to minimizing the *vehDelay*.

Besides, in the PSO algorithm, the population size M is set as 20; the dimensionality N is 33; the maximal number of iterations G is set as 1,000; ω is initialized as 0.9 [23] and linearly decreased to 0.4 during the search process; acceleration coefficients are set as that $c_1 = c_2 = 2.0$.

Each simulation lasts 100,000 time units. In the controllers, the initial duration Θ for each phase is set as 10, the threshold value Φ for extending the green phase is set as 5, the interval time Δ for recomputing UD is set as 10, and the length θ of all-red time is set as 5. The membership functions of FUZZY-TURN, FUZZY-JUMP, and FUZZY-MIX are set to be the same in [11] by human perception, whereas those of FUZZY-MIX-OPT are set according to the optimization results of the PSO algorithm.

B. *Results Comparisons on Number of Undetected Vehicles*

As an index of performance, the number of undetected vehicles during the simulation is shown in Table II. Vehicle actuated method has good performance in light and steady conditions C1-C4, but is inferior to the other four controllers in controlling conditions C5-C16. This is because the VA method adjusts the signal just according to whether there is at least one vehicle has been detected in the last time period. It employs little information of the roundabout and is thus limited in controlling heavy and complex traffic conditions. FUZZY-TURN and FUZZY-JUMP do better than VA. In addition, in the comparison of these two controllers, it can be observed that in controlling all the even numbered conditions, FUZZY-JUMP obtains less *vehMiss* than that obtained by FUZZY-TURN. As described in Subsection A, in even numbered conditions the differences between the north-south flow rates and the west-east flow rates are conspicuous. The FUZZY-JUMP always changes its green phase to the currently most urgent phase. The controller is much easier to switch the phase from one direction to the other direction than the FUZZY-TURN (which complies with predefined phase circle). Therefore, FUZZY-JUMP is more readily adapt to changing needs and circumstances and outperforms the FUZZY-TURN in controlling even numbered conditions. However, for some odd numbered conditions, the performance of FUZZY-TURN is better than FUZZY-JUMP. This may be because the FUZZY-TURN uses consistent adjacent phase sequences, and saves the cost of all-red time. When the urgency of phases between different directions is similar, the better real-timeness of FUZZY-JUMP is limited and the advantages of FUZZY-TURN emerge. But in general, FUZZY-JUMP obtains less *vehMiss* than FUZZY-TURN in most cases.

As a mixture of FUZZY-TURN and FUZZY-JUMP, the FUZZY-MIX controller outperforms FUZZY-TURN and FUZZY-JUMP for all the tested conditions. The reasons are summarized as follows. First, the inner layer of FUZZY-MIX uses consistent phase sequences, which helps to smooth the traffic flows and reduce congestion. Second, by the use of the outer layer, FUZZY-MIX is capable of breaking the phase sequences if the other direction of the roundabout is in more urgent than the current direction. By this way, the proposed controller can immediately respond to the current need of the roundabout. Third, compared to FUZZY-JUMP which considers the urgent degrees among the phases, the FUZZY-MIX concerns the urgent degrees among the phase subsets. The coarse-grained FUZZY-MIX controller does not change the phase sequences as frequent as the FUZZY-JUMP controller, and therefore induces less all-red time than the FUZZY-JUMP. The decrease in all-red time helps to improve the traffic efficiency of the roundabout. Last, when switching the phase from one phase subset to the other, the entrance phase of the subset is set as the phase which has maximum active flows. As the all-red time (before the entrance phase starts up) clears the vehicles in the circulatory lanes of the roundabout, the vehicles of the urgent direction can occupy the circulatory lanes as soon as possible. This mechanism helps to maximally relieve the traffic burden of the urgent direction.

In addition, by using PSO algorithm to optimize the membership functions adopted in FUZZY-MIX, the FUZZY-MIX-OPT controller further improves the performance of the proposed controller. These results also demonstrate that the traditional use of artificial membership functions does not provide the optimal performance of fuzzy systems. Therefore, using intelligent algorithms to optimize the membership functions of fuzzy logic controllers is promising. As shown in Table II, the FUZZY-MIX-OPT controller has the best performance among the five compared controllers. In controlling 14 out of the 16 tested conditions, the number of undetected vehicles obtained by FUZZY-MIX-OPT is 0. Only in controlling C14 and C16, the undetected vehicles still exist. This may due to that, in the two conditions, the flow rates increase to very big values, and the traffic volume approaches the capacity of the roundabout. In such cases, the PSO algorithm has little room to make further improvement for

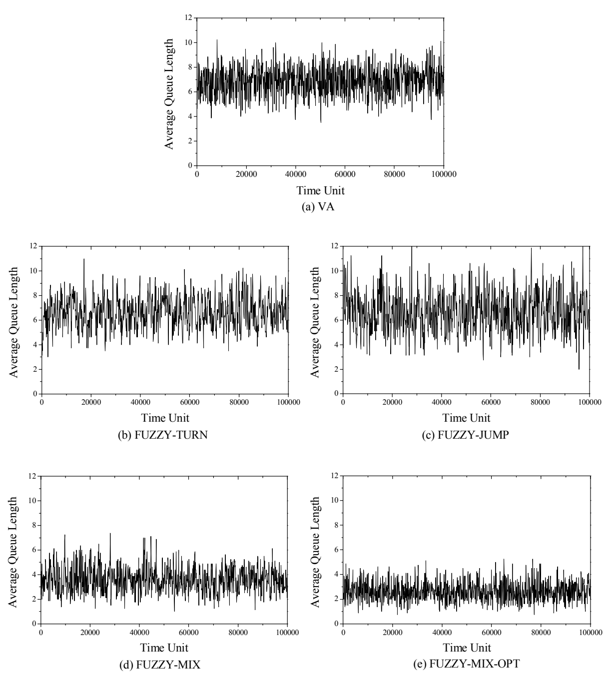

Figure 8. Average queue length during the simulation of condition C8.

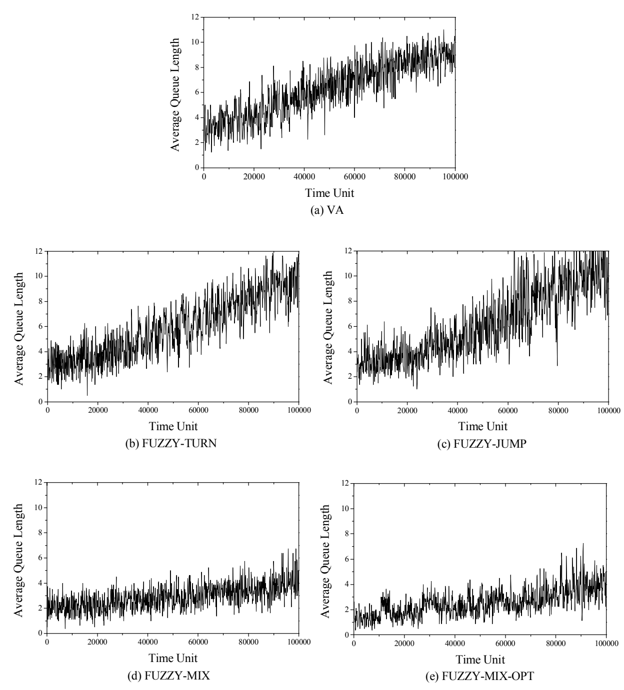

Figure 9. Average queue length during the simulation of condition C16.

reducing the undetected vehicles.

C. *Results Comparisons on Average Vehicle Delay*

In order to further compare the performance of the controllers, the average delay time of vehicles is computed, which is presented in Table III. The last column of Table III shows the mean of *vehDelay* among the 16 conditions. From Table III, it can be observed that VA has the largest vehicle delay, followed by FUZZY-JUMP and FUZZY-TURN. Our proposed FUZZY-MIX and FUZZY-MIX-OPT achieve the smallest delay. The mean of *vehDelay* obtained by FUZZY-MIX-OPT is 12.985 time units, which is about half of the value obtained by VA (28.969 time units), FUZZY-TURN (26.584 time units), and FUZZY-JUMP (26.642 time units). Moreover, compared with FUZZY-MIX with artificial membership functions (16.229 time units), the vehicle delay time of FUZZY-MIX-OPT decreases 20%. Therefore, the use of optimal membership functions trained by the PSO algorithm is very effective to improve the traffic efficiency of the roundabout.

In addition, the average length of the eight waiting-vehicle queues of the roundabout during the simulation is shown in Fig. 8 and Fig. 9, where steady condition C8 and time-varying condition C16 are taken as examples. It can be observed that the FUZZY-MIX-OPT controller has the shortest average queue length during the simulation, which further demonstrates the effectiveness and efficiency of the proposed optimal traffic signal controller.

VI. CONCLUSION

Traffic signal control is a crucial issue for modern roundabouts in recent years. Our preliminary work proposes a FUZZY-MIX controller which has two fuzzy layers hybridizing the FUZZY-TURN and FUZZY-JUMP controllers to signalize roundabouts. The paper focuses on applying PSO algorithm to optimize the membership functions of FUZZY-MIX. Therefore, a PSO-based FUZZY-MIX-OPT controller is developed, which adopts optimal membership functions instead of the previously used manual functions. Simulations are done on a roundabout with four approaches and two circulatory lanes. Simulation results show that the proposed FUZZY-MIX-OPT controller is very effective and can be regarded as a suggested scheme for the signal control of modern roundabouts.

REFERENCES

[1] C. W. Doh, *The principles of Transportation Engineering*. Seoul, Korea: Chong-Mun-Kak, 1989.

[2] J. W. Kim and B. M. Kim, "A GA-based fuzzy traffic simulation for crossroad management," in *Proc. 2001 IEEE Congr. Evol. Comput.*, vol. 2, pp. 1289-1295.

[3] P. G. Waingankar, U. Academy, and I. Nashik, "Fuzzy logic based traffic light controller," in *2007 IEEE Int. Conf. Ind. Inform. Syst.*, pp. 107-110.

[4] E. Azimirad, N. Pariz, and M. B. N Sistani, "A novel fuzzy model and control of single intersection at urban traffic network," *IEEE Syst. J.*, vol. 4, no. 1, pp. 107-111, Mar. 2010.

[5] J. H. Lee and H. Lee-Kwang, "Distributed and cooperative fuzzy controllers for traffic intersections group," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 2, pp. 263-271, May 1999.

[6] M. M. M. Fahmy, "An adaptive traffic signaling for roundabout with four-approach intersections based on fuzzy logic," *J. Comput. Inform. Technol.*, vol. 15, no. 1, pp. 33-45, 2007.

[7] W. Wei and M.-J. Wang, "Fuzzy-GA-based traffic signal control," in *Proc. 2nd Int. Conf. Mach. Learning Cybern.*, 2003, vol. 1, pp. 645-650.

[8] J. Qiao, N.-D. Yang, and J. Gao, "Two-stage fuzzy logic controller for signalized intersection," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no.1, pp.178-184, Jan. 2011.

[9] M. C. Choy, D. Srinivasan, and R. L. Cheu, "Cooperative, hybrid agent architecture for real-time traffic signal control," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 5, pp. 597-607, Sept. 2003.

[10] P. G. Balaji and D. Srinivasan, "Multi-Agent System in Urban Traffic Signal Control," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 43-51, Nov. 2010.

[11] Y.-J. Gong, J. Zhang, O. Liu, and Y. Li, "A novel fuzzy model for the traffic signal control of modern roundabouts," in *IEEE. Int. Cont. Syst. Man Cybern.*, 2011, pp. 1777 – 1782.

[12] J. Zhang, H. S. H. Chung and W. L. Lo, "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms", *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 326-335, Jun. 2007.

[13] J. Zhang, Z.-H. Zhan, Y. Lin, N. Chen, Y.-J. Gong, H. S. H. Chung, Y. Li, and Y.-H Shi, "Evolutionary computation meets machine learning: a survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp.68-75, Nov. 2011.

[14] W.-N. Chen and J. Zhang, "Ant colony optimization approach to grid workflow scheduling problem with various QoS requirements", *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 31, no. 1, pp. 29-43, Jan 2009.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE Int. Conf. Neural Netw.*, 1995, vol. 4, pp. 1942-1948.

[16] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, 1995, pp 39-43.

[17] Z.-H. Zhan, J. Zhang, and Y. Li, "Orthogonal learning particle swarm optimization", *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832-847, Dec 2011.

[18] Z.-H. Zhan, J. Zhang, Y. Li, and Henry Chung, "Adaptive particle swarm optimization", *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362-1381. Dec 2009.

[19] W.-N. Chen, J. Zhang, H. S. H. Chung, W.-L. Zhong, W.-G. Wu, and Y.-H. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems", *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278-300, Apr. 2010.

[20] M. R. AlRashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in electric power systems," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 913-918, Aug. 2009.

[21] J.-H. Zheng and M. Kamitani, "Evaluation of lost time in signal control for traffic flow analysis," in *Int. Symp. Inform. Eng. and Electron. Commerce*, 2009, pp. 463-466.

[22] L. Xue, L.-H. Lin, and R. Tian, Rui, "Researching on left-turning traffic organization and control at urban signalized intersections," in *Int. Conf. Optoelectronics Image Process.*, 2010, vol. 2, pp. 331-334.

[23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. 1998 IEEE Congr. Evol. Comput.*, pp.69-73.