Road-Junction Traffic Signal Timing Optimization by an adaptive Particle Swarm Algorithm

Juan Chen
Department of Control Science and Engineering
Tongji University
Shanghai, P. R. China
renqing1976@163.com

Lihong Xu
Department of Control Science and Engineering
Tongji University
Shanghai, P. R. China
xulhk@163.com

***Abstract*— The purpose of this paper is to investigate the application of Particle Swarm Optimization(PSO) Algorithm in solving the traffic signal timing optimization problem. A local fuzzy-logic controller (FLC) installed at each junction is used to provide some initial solutions for the particle swarm optimization algorithm. Coordination parameters from adjacent junctions are taken into consideration by local fuzzy-logic controller.The membership functions and the rules of the Fuzzy Logic Controller (FLC) are optimized using the standard Particle Swarm Optimization(SPSO) algorithm. A new particle swarm optimization algorithm is used to optimize the average delay and average number of stops for adjacent junctions and to handle the constraints. The simulation results show that the delay per vehicle can be substantially reduced under constant traffic demands and time-varying traffic demands, particularly when the traffic demands on the upstream is larger than the traffic demand on the downstream. The implementation of this method does not require complicated hardware, and such simplicity makes it a useful tool for offline studies or real-time control purposes.**

***Keywords*—Particle swarm optimization(PSO), fuzzy-logic controller (FLC),nonlinear optimization, road-traffic control, coordination**

I. INTRODUCTION

Road-Junction congestion is now a part of daily life in urban cites. The growing population worsens traffic congestion. Since less room is left for the city planners to device new measures to accommodate the ever-increasing road-traffic demand, effective and coordinated traffic control may be an essential tool for relieving the congestion to ensure the safe passage of vehicles, to minimize delay and to maximize junction utilization.

In order to maximize the throughout of the available infrastructure, traffic control should be imposed on the level of an area, rather than on an isolated junction. Traffic signal control mainly falls into two categories: fixed-time and traffic-responsive control.

AI techniques have found quite a number of application in transportation systems,such as [1],[2],[3], [5],[6],[7],[8],etc. These studies showed a certain degree of success, particularly when the traffic is not heavy and the turning percentage is small. However, when traffic is heavy, more phases have to be introduced to the cycle to cater to the turning traffic.In [4] a decentralized approach on road-junction traffic control is proposed, a local controller is installed at each junction and a dynamic programming(DP) technique is proposed to derive the green time for each phase in a traffic cycle. These works are capable of handling junctions with turning traffic. But it has been pointed out that DP method in [4] does not always produce the optimal control actions even though it improves the traffic control performance in general,and the computation time required by the DP technique takes nearly double of the time for fixed-offset control coordination, which make it not particularly efficient in large systems.

Swarm intelligence is based on the social behavior of flocks of birds/schools of fish and the success of the swarm due to the communication established between them. Particle Swarm Optimization (PSO) originally developed by [9],[10],which is a population-based algorithm. PSO is initialized with a population of candidate solutions. Each candidate solution in PSO, called particle, has associated a randomized velocity, moves through the search space. Each particle keeps track of its coordinates in the search space, which are associated with the best solution's fitness ,$pbest$,it has achieved so far. Another best value tracked by the global version of the particle swarm optimizer is the overall best value, $gbest$, and its location, obtained so far by any particle in the population. The PSO has been found to be robust and fast in solving nonlinear, non-differentiable, multimodal optimization problems.

We proposed a new particle swarm optimization algorithm to optimize the traffic control signal on road-junction traffic system in this study. Each junction is equipped with a local fuzzy-logic controller to allocate the green time for each phase in a cycle, the standard PSO algorithm is used to optimize the membership functions and the rules of the Fuzzy Logic Controller (FLC). Some initial solutions are provided for the new particle swarm optimization algorithm. The cycle length , the phasing splitting and offset of all signals in the adjacent junctions system are optimized using the new PSO algorithm.

II. LOCAL CONTROLLER

A. Fuzzy logic controller

In order to accommodate turning traffic at a junction, a complete control cycle should consist of a minimum of four

1–4244–0342–1/06/$20.00 © 2006 IEEE

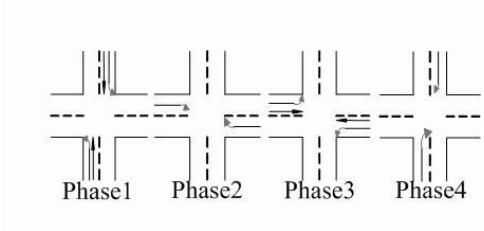

Fig. 1. Phasing setting of a junction with right turnings.

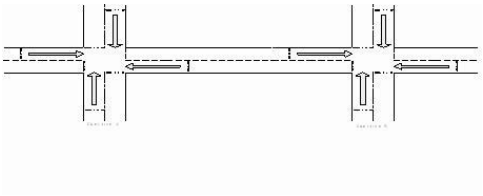

Fig. 2. Junction Setup

phases , as shown in Fig.1, allowing traffic from the four phases ($\Phi_i, i = 1, 2, 3, 4$) to pass through the junction in turn.

The following assumptions are made in this junction setup.

1) all vehicles are right-hand driven, hence left-turning traffic does not block the traffic of the opposite direction.
2) It is assumed that the flow rates among the traffic streams is independent of each other.
3) The minimum effective green time for each phase is 0s, the maximum effective green time for each phase is 120s.
4) The order of the traffic phases within a cycle is well defined and fixed.
5) Vehicle arrivals are subject to uniform distribution and the mean equal to the flow rate.
6) Saturation flow for each arm at the junction is d vehicles per second.
7) The number of vehicles that the junction can hold at its arms is not bounded in this study.
8) The traffic at one arm is collectively regarded as one stream of lane, and the discharge rate of vehicles is the same as the saturation flow.

The objective of the controller is to determine the effective green times t_g for each phase, in such a way that the total delay at the junction is minimized.

Assuming that traffic phase Φ_j is given the right-of-way, the queue lengths at the traffic phase Φ_i are:

$$q_i^{'} = \begin{cases} q_i + f_i T_{lost} + f_i t_g - dt_g & i = j \quad (1a) \\ q_i + f_i T_{lost} + f_i t_g & i \neq j \quad (1b) \end{cases}$$

Where q_i and $q_i^{'}$ are the queue lengths at traffic phase Φ_i before and after t_g, the green time given to Φ_j, respectively. As there are two queues in opposite directions in a traffic stream, the longer queue between the two is taken as q_i or $q_i^{'}$. f_i is the flow rate of Φ_i, T_{lost} is the lost time.

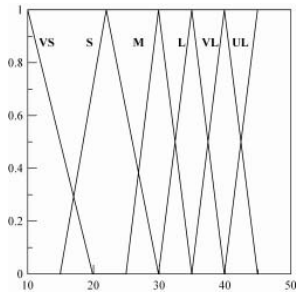

Fig. 3. Queue length.

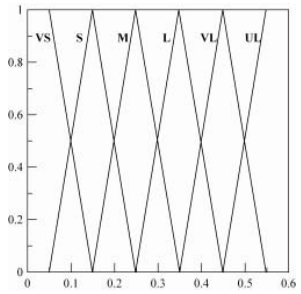

Fig. 4. Flow rate.

As attempting to attain t_g directly requires substantial computational time because of the number of inputs to the control problem. In order to simplify the control for real-time applications, a two-level FLC has been developed, in which an estimate of the green time is first made according to (1a), adjusted by a correction with (1b), so that

$$t_g = t_{gg} + t_{gh}$$

The first level of the FLC is a two-input, single-output controller, in which q_j and f_j are the input. This level provides an estimation of the green time required t_{gg} to clear the queue accumulated, just before the green signal is applied.

The second level of the controller determines an adjustment t_{gh} according to the relationship of the flow rates among the traffic phases. (i.e., f_j and $\sum_{i=1}^{4} f_i$)

The structure divides a three-input (q_j,f_j,$\sum f_i$), one-output (t_g) FLC into two levels of control, with a two-input, one-output FLC at each level. Fig.3 , Fig.4 and Fig.5 illustrate the membership functions of the two input variables of the first and second level of the FLC, respectively.

With the two FLCs, the total number of rules in the controller is reduced from N^3 to $2N^2$, where N is number of fuzzy labels of each input. The center-of-area method has been adopted for defuzzification.

B. Particle swarm optimization learning

In this study, SPSO is used to determine the optimal input and output membership functions and the optimal rules for the local fuzzy logic controllers.Fig.6 shows the block diagram of FLC with PSO learning.

The objective of the PSO learning algorithm are to minimize average delay D of vehicles in all traffic streams (resource

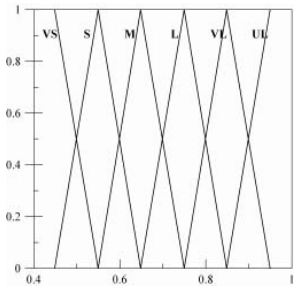

Fig. 5. Total flow rate.

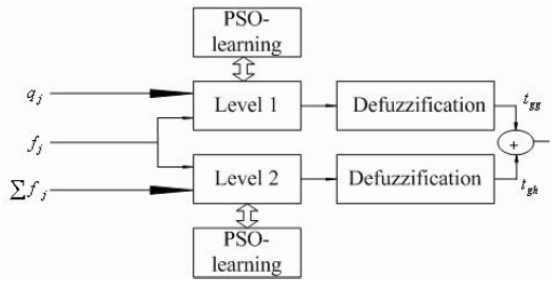

Fig. 6. Block diagram of the FLC with PSO learning

allocation) and to maximize the number of vehicles C leaving the junction(resource utilization). The fitness function $F = C - D$ is defined and the standard PSO(SPSO) is used to maximize the fitness function.

The SPSO algorithm is described in the following:

1) Initialize a population of particles with random positions and velocities in the n-dimensional problem space:

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$$

$$P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$$

$$V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$$

2) For each particle, evaluate its fitness value.

3) Compare each particles fitness evaluation with the current particles $pbest$. If current value is better than $pbest$, set its $pbest$ value to the current value and the $pbest$ location to the current location in n-dimensional space.

4) Compare fitness evaluation with the populations overall previous best. If current value is better than $gbest$, then reset $gbest$ to the current particles array index and value.

5) Change the velocity and position of the particle according to the following equations respectively :

$$V_{id} = (\omega \times V_{id} + c_1 rand()(P_{id} - X_{id}) + c_2 rand()(P_{id} - X_{id})) \quad (2)$$

$$X_{id} = (X_{id} + V_{id}) \quad (3)$$

6) Loop to step (2) until a stopping criterion is met, usually a maximum number of iterations (generations).

The vector $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ stands for the potential solution to a problem in D-dimensional space, with particle i, $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$ stands for the velocity of the ith particle and $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ represents the best previous position (the position giving the best fitness value) of the ith particle. The index g represents the index of the best particle among all the particles in the swarm. Variable ω is the inertia weight, c_1 and c_2 are positive constants; $rand()$ is random numbers in the range $[0, 1]$ generated according to a uniform probability distribution. Particles velocities along each dimension are clamped to a maximum velocity V_{max}. If the sum of accelerations causes the velocity on that dimension to exceed V_{max}, which is a parameter specified by the user, then the velocity on that dimension is limited to V_{max}.

The inertia weight ω represents the degree of the momentum of the particles. The second part is the cognition part, which represents the independent behavior of the particle itself. The third part is the social part, which represents the collaboration among the particles. The constants c_1 and c_2 represent the weighting of the cognition and social parts that pull each particle toward pbest and gbest positions.

III. PSO Optimization

The Particle Swarm Optimization (PSO) algorithm has become increasingly popular in the last few years, mainly in numerical optimization tasks [9]. The particle swarm is trying to perform as a self-organizing system with extraordinarily flexibility [11]. To solving the real-world problem, two issues should be handled:(a)to handle the constraints and (b)to keep the diversity of the particle.

The constraint handling methods [12],[13],[14],[19],[20], [15],[16],[17],are employed for constructing the fitness landscape $F(\vec{x})$. To avoiding the laborious and often troublesome setting of the penalty coefficients in the methods based on penalty functions [16], the handling methods that following Debs criteria [12] is often applied [20],[18]. However, to satisfying boundary constraints, the conventional handling methods [20],[11], seem not always suitable for the efficient flying of particle swarm. another problem is that although the method that following Deb's criteria is applied on swarm algorithm for the problem with inequality constraints successfully [20],[18],it is still difficult to deal with equality constraints [19],epically when the violation values are very small [18].

Work presented in [23] describes the complex task of parameter selection in the PSO model. PSO has been proved to be a competitor to the standard genetic algorithm (GA). Comparisons between PSO and GA were done with regards to performance by [21], which points out that the PSO performs well in the early iterations, but has problems in reaching a near optimal solution in several benchmark functions.

To overcome this problem, some researchers have employed methods with adaptive parameters [23]. One is deterministic mode, which the PSO parameters are changed according to the deterministic rules, such as a linear decreased inertia weight as the number of generation increasing [23], which are obtained according to the experience. The other is adaptive mode, which

adjusts the parameters according to the feedback information, such as fuzzy adaptive inertia weight [22]. However, currently, we still have not captured the relations between different parameters and their effects toward different problem instances, such as dynamic optimization problems [24], due to the complex relationship among parameters.

We have discussed a SPSO algorithm in the above section,in order to handle constraints and to improve the average performance of the PSO, a new PSO algorithm is proposed, which is similar to SPSO except for three aspects: the generation of the initial solutions, the way to taking off inactive particles, and the mechanism adopted to handle the constraints.

1) Generation of initial solutions

Firstly,some initial solutions are provided by the local controller described in the previous section to utilize the nature information of the system according to the following methods.

A traffic cycle at a junction is considered as a number of consecutive stages. At each stage, the traffic condition at the junction is described collectively by a state. suppose that the four phases in one traffic-light cycle at a junction is in order. Decisions must be made for the traffic at the junction to move on from one stage to another. The FLC described in the previous section is adopted as local junction controller and it determines an effective green time t_g for each stage.In this paper, the cycle time is predefined according to different flow rates, the offset of junctions having common cycle length is the traveling time of the vehicle in the common link. Junction coordination is introduced by adding an adjustment parameter t_c, which can be either positive or negative. Here we denote t_c by the following set:

$$t_c \in \{-15, -10, -5, 0, 5, 10, 15, 20\}(s)$$

let t_{g1} denote the green-time allocation for the first stage in a cycle defined by the FLC, updating t_{g1} according to the following equations:

$$t_{g1} = t_{g1} + t_c$$

the green-time allocation for the rest three stages in a cycle is updated according to the following equations:

$$t_{gi} = t_{gi} + t_c \times \frac{q_i}{\sum_{j=1}^{4} q_j}, i = 2, 3, 4$$

with a finite set of t_g,we get 8 kinds of allocation of the green-time for every stage in a cycle.

Given 8 kinds of allocation results for a single junction, we get $8 \times 8 = 64$ kinds of allocation mechanism for a two junctions traffic system. The first 20 kinds of allocation mechanism is selected as the initial solutions for the PSO algorithm according to the control objective of the whole system .

Secondly,some initial solutions are generated randomly to keep diversity of the particle.

2) Mechanism to handle inactive particles

As evolution goes on, some particles become inactively while lost both of the global and local search capability in the next generations. Hence, the evolution process will be stagnated. If $gbest$ is located in a local optimum, then the swarm becomes premature convergence as all the particles become inactively. It is necessary to recognize and replace the inactive particle by a fresh one adaptively so as to stimulating the swarm with sustainable development by maintaining the social diversity of swarm.

Taking the problem information into account can improve the algorithm performance. For each timing design mechanism, there exists process deviation σ_d for each parameter x_d. For a given point $\vec{o}$, its similar set S_s is defined as all the points within the range of the deviations. Then for $\forall \vec{x} \in S_s$, which $x_d - o_d \leq \sigma_d$ come into existence in all dimensions, is noted as $\tilde{x} \cong \tilde{o}$, i.e. $\tilde{x}$ is similar to $\tilde{o}$. Then a particle can be identified as inactive if it is always flying in S_s of $\vec{p}_g$ and without any improvement on its fitness in the following T_c generations.

The pseudo code for recognizing inactive particle is shown below [11], which is executed after equations (2,3). The SC array is employed to store the times that are similar to the $gbest$ in successively for each particle. T_c is a predefined constant. It need not be too large since the swarm dynamics is dissipative. In this paper, T_c is set as 5,σ_d is set as 0.01.

$$IF(\vec{x}_i \cong \vec{g}_i and i \neq g)//similar to gbest?$$

$$Then SC[i] + +;$$

$$else SC[i] = 0;$$

$$IF(SC[i] > T_c)$$

$$Then the ith particle is inactive$$

$$The pseudo code$$

If the ith particle is recognized as inactive, then it is replaced by a fresh particle, which the location $\vec{x}_i$ and $\vec{\nu}_i$ is reinitialized at random, and $SC[i]$ is reset to 0.

3) Mechanism to Handle Constraints

3.1) Fitness function design

Typical traffic signal timing optimization problem can be defined as:

$$Minimize : f(\vec{x})(MINTYPE)$$

$$subject to : g_j(\vec{x}) \leq 0 (1 \leq j \leq m_0, j \in Z)(CONTYPE)$$

$$h_j(\vec{x}) = 0 (m_0 + 1 \leq j \leq m, j \in Z)(CONTYPE)$$

where $\vec{x} = (x_1, \cdots, x_d, \cdots, x_D)$ and the search space S is a D-dimensional space bounded by all the parametric constraints $x_d \in [l_d, u_d]$. f is objective function, g_j and h_j are inequality and equality constraints, respectively. The set of points, which satisfying all the constraints, is denoted as feasible space S_F, and then the infeasible space is $S_I = \bar{S}_F \bigcap S$.

The total fitness function $F(\vec{x})$ is defined as:

$$F(\vec{x}) = < F_{OBJ}(\vec{x}), F_{CON}(\vec{x}) >$$

where $F_{OBJ}(\vec{x}) = f(\vec{x})$ and $F_{con}(\vec{x}) = \sum_{j=1}^{m} r_j G_j(\vec{x})$ are the fitness functions for objective function and constraints, respectively. Here $G_j(\vec{x}) = max(0, g_j(\vec{x}))$, and r_j are positive weight factors, which default value is equal to 1.

3.2) Mechanism to Handle boundary Constraints

The boundary constraint-handling method is following Debs criteria [20],[18]: a) any feasible solution is preferred to any infeasible solution; b) among two feasible solutions, the one having better objective function value is preferred; c) among two infeasible solutions, the one having smaller constraint violation is preferred.

The boundary constraints are handled using Periodic mode [11]. For each point $\vec{x}_i$, its fitness will be calculated by a mapping point $\vec{z}_i = (z_{i1}, \cdots, z_{id}, \cdots, z_{iD})$, which for dth dimension, it has:

$$\tilde{M}_P(x_d \longrightarrow z_d) : \{ \begin{array}{cc} z_d = u_d - mod((l_d - x_d), s_d) & IFx_d < l_d \\ z_d = l_d + mod((x_d - u_d), s_d) & IFx_d > u_d \\ z_d = x_d & IFx_d \in [l_d, u_d] \end{array}$$

where mod is the modulus operator,$s_d = u_d - l_d$ is the parameter range of the dth dimension. The ultimate optimized point $\vec{x}^*$ is calculated from $\tilde{M}_P(gbest \longrightarrow \vec{x}^*)$, which is satisfying the boundary constraints.

Comparing with conventional boundary constraints handling methods, the Periodic mode have some advantages to enhance robustness of the particle swarm[25].

3.3) Mechanism to Handle equality Constraints

3.3.1)basic constraint handling(BCH)[25]

The basic constraints handling (BCH) rule for goodness evaluation is realized by comparing any two points $\vec{x}_A$,$\vec{x}_B$, $F(\vec{x}_A) \leq F(\vec{x}_B)$when

$$\{ \begin{array}{ll} F_{CON}(\vec{x}_A) < F_{CON}(\vec{x}_B) & OR \\ F_{OBJ}(\vec{x}_A) \leq F_{OBJ}(\vec{x}_B), & F_{CON}(\vec{x}_A) = F_{CON}(\vec{x}_B) \end{array} \quad (4)$$

The BCH rule is following the Deb's criteria [6]: a) any $\vec{x} \in S_F$ is preferred to any $\vec{x}! \in S_F$; b) among two feasible solutions, the one having better F_{OBJ} is preferred; c) among two infeasible solutions, the one having smaller F_{CON} is preferred.

3.3.2) Adaptive Constraint Relaxing(ACR)[25]

In this paper, the fitness is transformed by constraints relaxing rule so as to match swarm algorithms.

An additional rule is applied on equation (4):

$$F_{CON}(\vec{x}) = max(\varepsilon_R, F_{CON}(\vec{x})) \quad (5)$$

It means that among two points in $S_F^{'}$, the one with smaller F_{OBJ} is preferred. The adjusting of ε_R^t is referring to a data repository $\underline{P} = \vec{p}^t | 1 \leq i \leq N, i \in Z$, which is updated frequently,normally. In $\underline{P}$, the number of points with $F_{CON}(\vec{p}^t) > \varepsilon_R^t$ is defined as $N_\varepsilon^{(t)}$.

Initially, the $\varepsilon_R^{(0)}$is set as maximum F_{CON} value in K_o. Then the adaptive relaxing (ACR) rule is employed for ensuring $\varepsilon_R^t \longrightarrow 0$, by the following combined rules: a) the basic ratio-keeping sub-rules; b) the forcing sub-rule.

The basic ratio-keeping sub-rules try to keep a balance between the points inside and outside the $S_F^{'}$:

$$\begin{aligned} &IF(r_N^{(t)} \leq r_l)THEN\varepsilon_R^{(t+1)} = \beta_u \varepsilon_R^{(t)} \\ &IF(r_N^{(t)} \geq r_u)THEN\varepsilon_R^{(t+1)} = \beta_l \varepsilon_R^{(t)} \end{aligned} \quad (6)$$

where $r_N^{(t)} = \frac{N^{(t)}}{\varepsilon}$ is the ratio of the points inside the $S_F^{'}$,$0 \leq r_l < r_u \leq 1$,$0 < \beta_t < 1 < \beta_u$.

The $\varepsilon_R^{(t)}$ is not varied as $r_N^{(t)} \in (r_l, r_u)$. If the $\varepsilon_R^{(t)}$ is large at the end of learning cycles, then $\vec{g}^{(T)}$ may even not enter the S_O. Hence it is necessarily to prevent the $\varepsilon_R^{(t)}$ from stagnating for many learning cycles, especially for the cases that the elements in K_O are hardly to be changed.

The forcing sub-rule forces the $\varepsilon_R^{(t)}|_{t \longrightarrow T} \longrightarrow 0$:

$$IF(t \geq t_m)THEN\varepsilon_R^{(t+1)} = \beta_f \varepsilon_R^{(t)} \quad (7)$$

where $0 < \beta_f < 1$ and $0 < t_m < T$.

The default parameter values include: $r_l = 0.25$,$r_l = 0.75$, $\beta_f = \beta_l = 0.618$,$\beta_u = 1.382$,$t_{th} = 0.5T$.

IV. Model Formulation

The network performance index PI is a function of signal setting variables $\Psi = (c, \theta, \Phi)$. The objective function is therefore to minimis PI subject to signal setting constraints. This gives the following minimization problem:

$$Minimise_{\Psi \in \Omega_0} PI(\Psi) = \sum_{a \in L} Q(\Psi) + W(\Psi)$$

where $Q(\Psi)$ is the average delay per vehicle at the two junctions, $W(\Psi)$ is the average number of stops per vehicle at the two junctions. subject to

$$\Psi(c, \theta, \Phi) \in \Omega_0$$

cycle time constraints,

$$c_{min} \leq c \leq c_{max}$$

values of offset constraints

$$0 \leq \theta \leq c_{max}$$

green time constraints,

$$\Phi_{min} \leq \Phi \leq \Phi_{max}$$

$$\sum_{i=1}^{m} (\Phi_i + I_i) = c, \forall m \in M, \forall n \in N$$

The signal timing constraints are given as follows:

Common network cycle time:

$$c_{min}, c_{max} = 36, 120s$$

Offset values:

$$\theta_{min}, \theta_{max} = 0, 120s$$

Minimum green time for signal stages:

$$\Psi_{min} = 7s$$

Intergreen time between the stages:

$$I_{1-2}, I_{2-1} = 5s$$

TABLE I

DELAY IMPROVEMENT BY THE PSO ALGORITHM

0.7	0.8	0.9	A-B
-9.3	-2.7	45.4	0.5
19.2	38.2	84.6	0.6
-58.9	59.7	47.9	0.7

V. SIMULATION RESULTS

To demonstrate the performance of the PSO optimization for junction coordination, it is applied on a simple two-junction network which is subject to various traffic demands,and the FLCs presented are adopted as the local junction controllers. The performance is compared with that obtained by two local FLCs with fixed offsets.

A. Junction setup:

A simple network of two connecting junctions is used for testing, and its layout is shown in Fig.2. Vehicle detectors are installed in front of and in the vicinity of the junctions.There are shared between the two junctions. The connecting link between the two junctions can hold up to 50 vehicles in each direction, and the vehicles are moving in , and no dispersion is assumed.The turning percentage (both left and right) are set to 10 percentage.

B. Constant traffic demands:

The network is first tested with constant traffic flow rates feeding the incoming roads of the network.For comparison, two local FLCs are employed at the two junctions, while they are coordinated by a fixed offset. The offset is set as the average travel time between the two junctions, and the value is kept unchanged.

Different combinations of flow rates (in vehicle per second) are imposed on junction A and B.The saturation flow at the two junctions A and B is 1 vehicle/second. The percentage reduction in delays under different traffic demands by the coordinated optimization is summarized in table.1.

In this study,several algorithm settings are tested. For adaptive PSO, ω is fixed as 0.4, the number of particles N is set as 40 for all PSO versions.we had 30 trial runs for every instance.

Fig.7 shows the objective value(average delay) during 200 generations for different algorithm. It shows that the original PSO version with w=1, performed worst in all cases since it is worked in chaotic state. The PSO version with w=0.4 converged fastly and stagnated at the last stage of evolution, since it is worked in dissipative state. The PSO version with a linearly decreasing w which from 0.9 to 0.4 performed better than the PSO version with w=0.4 at last which converged slowly at the early stage while fastly at the last stage,the PSO with ACR constraints handling mechanism performed better than PSO with BCH constraints handling mechanism. Moreover, the PSO with ACR constraints handling mechanism performed as the best in all cases, which evolving sustainable

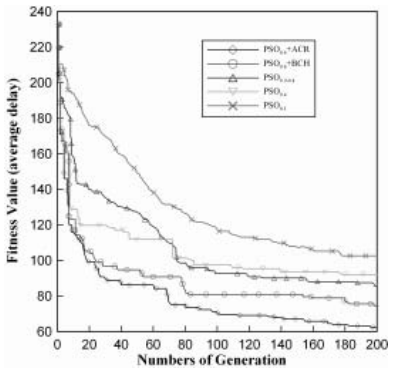

Fig. 7. Convergence of PSO optimizer

TABLE II

DELAY COMPARISON OF PSO ALGORITHM AND FIXED-OFFSET CONTROL

	fixed offset	PSO	reduction
NT	25.49	26.78	-5.06
MP	147.30	98.61	33.05
DT	77.95	83.42	-7.02
EP	115.460	73.90	36.00

when the evolution of PSO with same w is going to be stagnated.

C. Time-Varying traffic demands

Under this test, the same two-junction network is subject to a 24-h flow rate profile as shown in Fig.8(in [4]), which resembles the traffic demands throughout a day at a typical road junction in a busy city with four time zones, namely night-time,NT", morning peak "MP", Daytime "DT", and evening peak "EP",

The delay variation is given in Fig.9 and the average delays in the four time zones are summarized in Table.2.

From the results, the PSO optimization performs better in time-varying conditions, epically under the peak periods.It achieves more than 26 percentage reduction in the average delay and the response to changes of flow rates,such as overshoot and transient, is significantly superior. On the other

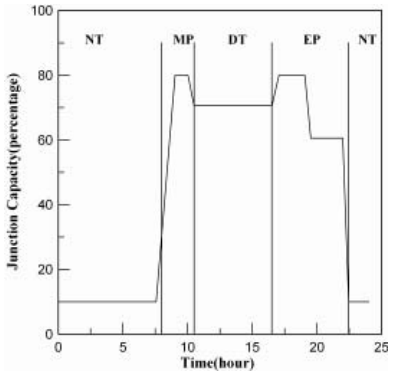

Fig. 8. Flow-rate pattern for the continuous flow rate changeling test

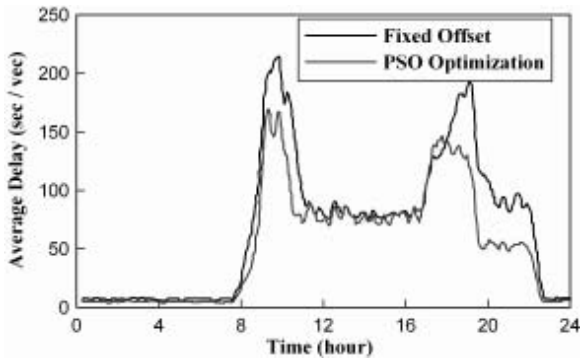

Fig. 9. Average delays by PSO OPTIMIZATION and fixed-offset coordination under time-varying traffic demands

hand, the fixed-offset coordination performs slightly better than the PSO periods, the flow rates of the two junctions are similar. In general, the PSO coordination provides a flexible means to deal with the imbalanced and time-varying traffic demands on the junctions, which usually happens in real life. It significantly reduces delays on the average, and alleviates the momentary buildup of queues.

It has to be pointed out that the convergence rate of our PSO algorithm is very high. Compared to the DP method in [4], the PSO algorithm not only get better performance(in some cases), but also need less computation time than DP method in [4].

VI. Conclusion

In this paper, a local fuzzy-logic controller is used to handle the local control process, a SPSO is used to optimize the scales of the FLC. The local controllers then provides some initial solutions for adaptive PSO , then the adaptive PSO algorithm is used to implement the optimization process. The methods presented here does not limit the number of junctions in the network. The local controllers are physically and functionally independent from each other. The computational effort is distributed to the local controllers.

It has been shown in this paper that it is possible to optimize the input and output membership functions and the rules of fuzzy systems using PSO algorithm. In the adaptive PSO algorithm, a mechanism to recognize and replace inactive particles is applied to keep the diversity of the particles, boundary and equality constraints are handled using periodic mode and BCH, ACR rules,respectively.

The results in this paper show it is possible to carry out traffic signal timing design problem using PSO algorithm under different traffic demands. In addition,it shows that the PSO can find a much better spread of optimal signal design plans with high convergence speed.Further work will address the implementation of PSO in network traffic signal control system to simultaneously optimize multiple objectives and parameters for a more complex junction than a simple cross junction.

References

[1] I. Pavlidis, V. Morellas, and N. Papanikolopoulos, "A vehicle occupant counting system based on near-infrared phenomenology and fuzzy neural classification," IEEE Trans. Intell. Transp. Syst., vol. 1, no. 2, pp. 72-85, Jun. 2000.

[2] L. Zhao and C. E. Thrope, "Stereo- and neural network-based pedestrian detection," IEEE Trans. Intell. Transp. Syst., vol. 1, no. 3, pp. 148-154, Sep. 2000.

[3] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic license plate recognition," IEEE Trans. Intell. Transp. Syst., vol. 5, no. 1, pp. 42-53, Mar. 2004.

[4] T. H. Heung, T. K. Ho, Y. F. Fung, Coordinated road-junction traffic control by dynamic programming, IEEE Trans. Intell. Transp. Syst., vol. 6, no. 3, pp. 341C350, Sep. 2005.

[5] C. R. Reeves, Genetic Algorithms: Principle and Perspective: A Guide to GA Theory. Boston, MA: Kluwer, 2003.

[6] M.D.Foy R.F.Benekohal, D.E.Goldberg,Signal Timing Determination Using Genetic Algorithms. In Transportation Research Record 1365, TRB. Washington, DC, 1992,pp.108-115

[7] M.A.Hadi, C.E.Wallace, Hybrid Genetic Algorithm To Optimize Signal Phasing and Timing. In Transportation Research Record 1421, TRB, Washington, DC,1993, pp.104-112

[8] G.Q.Memon, A.G.R.Bullen, Multivariate Optimization strategies for realtime Traffic Control Signals.Presented at the 75th Annual Meeting of the Transportation Board, Washington,D.C.,(1996)

[9] J. Kennedy and R. C. Eberhart. Swarm Intelligence. Morgan Kaufmann, UK, 2001.

[10] R.C. Eberhart and J.Kennedy A New Optimizer Using Particle Swarm Theory in proc. of sixth International Symposium On Micro Machine and Human Science (Nagoya, Japan) , IEEE Service Center 1995, pp 39-43

[11] X.F.Xie, W.J.Zhang, Z.L.Yang. A dissipative particle swarm optimization. Congress on Evolutionary Computation, Hawaii, USA, 2002: 1456-1461

[12] K.Deb, An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311-338

[13] R.Farmani, J.A.Wright, Self-adaptive fitness formulation for constrained optimization. IEEE Trans. on Evolutionary Computation, 2003, 7(5): 445-455

[14] S.B.Hamida,M.Schoenauer, ASCHEA: new results using adaptive segregational constraint handling. Congress on Evolutionary Computation, Hawaii, USA, 2002: 884-889

[15] Z.Michalewicz,M.Schoenauer, Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation. 1996, 4 (1): 1-32

[16] K.E.Parsopoulos,M.N.Vrahatis, Particle swarm optimization method for constrained optimization problems. In Sincak, P., Vascak, J., Kvasnicka, V., Pospichal, J. (Eds.): Intelligent Technologies - Theory and Applications: New Trends in Intelligent Technologies, IOS Press, 2002: 214-220

[17] T.Ray, K.M.Liew, Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans. Evolutionary Computation 2003, 7(4): 386-396

[18] W.J.Zhang, X.F.Xie, DEPSO: hybrid particle swarm with differential evolution operator. IEEE Int. Conf. on Systems, Man and Cybernetics, Washington DC, USA, 2003: 3816- 3821

[19] X.Hu, R.C.Eberhart, Y.H.Shi, Engineering optimization with particle swarm. IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 2003: 53-57

[20] J.Lampinen, A constraint handling approach for the differential evolution algorithm. Congress on Evolutionary Computation, 2002: 1468-1473

[21] P. J. Angeline. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. Proc. of 7th Annual Conf. on Evolutionary Programming, 1998: 601-610 Y. Shi, R.

[22] Y. Shi, R. Eberhart. Fuzzy adaptive particle swarm optimization. IEEE Int. Conf. on Evolutionary Computation, 2001: 101-106

[23] Y. Shi, R. Eberhart. Emperical study of particle swarm optimization, Proc. of Congremm on Evolutionary Computation,1999;1945-1950

[24] R. Eberhart, Y. Shi. Particle swarm optimization: developments, applications and resources. IEEE Int. Conf. on Evolutionary Computation, 2001: 81-86

[25] X. F. Xie, W. J. Zhang. SWAF: swarm algorithm framework for numerical optimization. Genetic and Evolutionary Computation Conference (GECCO), Part I (LNCS 3102), Washington, USA, 2004: 238-250