Appl Intell (2016) 44:208–231
DOI 10.1007/s10489-015-0701-y

A swarm intelligent method for traffic light scheduling: application to real urban traffic networks

Wenbin Hu[1] · Huan Wang[1] · Liping Yan[1] · Bo Du[1]

Published online: 5 August 2015

Abstract Traffic lights play an important role nowadays for solving complex and serious urban traffic problems. How to optimize the schedule of hundreds of traffic lights has become a challenging and exciting problem. This paper proposes an inner and outer cellular automaton mechanism combined with particle swarm optimization (IOCA-PSO) method to achieve a dynamic and real-time optimization scheduling of urban traffic lights. The IOCA-PSO method includes the inner cellular model (ICM), the outer cellular model (OCM), and the fitness function. Our work can be divided into following parts: (1) Concise basic transition rules and affiliated transition rules are proposed in ICM, which can help the proposed phase cycle planning (PCP) algorithm achieve a globally sophisticated scheduling and offer effective solutions for different traffic problems; (2) Benefited from the combination of cellular automaton (CA) and particle swarm optimization (PSO), the proposed inner and outer cellular PSO (IOPSO) algorithm in OCM offers a strong search ability to find out the optimal timing control; (3) The proposed fitness function can evaluate and conduct the optimization of traffic lights' scheduling dynamically for different aims by adjusting parameters. Extensive experiments show that, compared with the PSO method, the genetic algorithm method and the RANDOM method in real cases, IOCA-PSO presents distinct improvements under different traffic conditions, which shows a high adaptability of the proposed method in urban traffic network scales under different traffic flow states, intersection numbers, and vehicle numbers.

Keywords Traffic lights · Scheduling · Optimization · Particle swarm optimization · Cellular automaton

1 Introduction

Traffic congestion in urban traffic networks is a costly problem that has brought out many severe problems, such as pollution, parking, and security. These problems have already restricted the urban development. In large cities, the possibility of the construction and expansion of urban roads would be constrained by the urban space more and more seriously. Even the simple idea that the government rebuilds the existing urban infrastructures in a reasonable planning may still be very hard to implement in practice. It has been widely recognized that the optimization scheduling of traffic lights at urban intersections provides an effective and economic way to solve traffic congestion problems [1–3]. The key behind the problem is to achieve a global optimal scheduling of traffic lights, which improves the urban traffic conditions comprehensively and accelerates the traffic flow through cities. Another important point desiring attention is that many cities have hundreds of intersections under the traffic light control. The behaviors of traffic lights at different intersections must be coordinated to achieve a common goal of optimizing traffic flow. The scheduling of hundreds of traffic lights is very complex and challenging.

In this paper, a dynamic and real-time traffic light optimization method based on cellular automata (CA) and particle swarm optimization (PSO) is proposed. The main motivations can be summarized as follows. (1) CA is a decentralized computing model, which provides an excellent

✉ Wenbin Hu
hwb@whu.edu.cn

1 School of Computer, Wuhan University, Wuhan, Hubei Province, China

platform for performing complex evolution with the help of only local information. Due to its simplicity, flexibility and efficiency in computer simulations, CA has been a powerful tool to conduct exploitations in physics, biology, social science, computer science, and so on [9, 10]. (2) Because of the faster computing speed and the better global searching ability in dynamic multi-objective optimization environment, the PSO become one of the most important areas in the field of swarm intelligence [11–14]. Actually, it has been applied to solve many complex optimization problems, but only a little previous literatures focus on optimizing the scheduling of urban traffic lights [16–19]. Meanwhile, in order to enhance the performance of PSO, many attempts have been made, such as studying particle's neighbor topologies [23, 24], applying comprehensive learning strategies [25], employing cooperative behaviors [26], considering time variant inertia and acceleration coefficients [27]. These work suggests that PSO have a huge potential to optimize the scheduling of urban traffic lights.

Based on the above considerations, an inner and outer CA mechanism combined PSO (IOCA-PSO) method is proposed in the paper, which aims to achieve a globally optimal scheduling of traffic flow by coordinating the traffic lights. The IOCA-PSO method consists of the inner cellular model (ICM), the outer cellular model (OCM), and the fitness function. ICM conserves the scheduling settings of traffic lights and provides a flexible way to control the change of traffic lights. OCM devotes into optimizing the setting in ICM. The fitness function is the link between ICM and OCM by calculating the fitness value, which is used to evaluate the settings in ICM and conducts the optimization in OCM. The main contributions of this paper lie on the following points.

(1) The IOCA-PSO method is proposed to optimize the global scheduling of traffic lights in the extensive urban area, which can efficiently control urban traffic lights dynamically and in real time. It can also achieve a comprehensive scheduling of all traffic lights, which includes the timing control, the phase sequence control, and the special phase controls for different kinds of traffic problems.
(2) In ICM, a phase cycle planning (PCP) algorithm is proposed, which can achieve the globally sophisticated control by the concise rules. The rules include the basic transition rules and the affiliated transition rules. Benefited from the discrete characters of CA, the basic transition rules determine the phase shifting control of traffic lights. The affiliated transition rules achieve the special phase controls for different kinds of complex traffic problems.
(3) In OCM, an inner and outer cellular PSO (IOPSO) algorithm is proposed based on CA and PSO. Deriving from a powerful CA-based mechanism that guides the site change of particles in OCM, the IOPSO algorithm achieves a better balance of the local exploitation and the global exploration. And the IOPSO algorithm helps OCM find the optimal timing scheduling in ICM in the limited time effectively.

The remainder of this paper are organized as follows. Section 2 summarizes the latest and influential work related to the traffic light scheduling. Section 3 details the problem of traffic light scheduling. Then the IOCA-PSO method is presented in Section 4. And Section 5 details the experiments and discussions. Finally, conclusions and perspectives are concluded in Section 6

2 Literature review

Researchers have attempted to optimize the traffic light scheduling by a variety of methods. Most methods can be divided into six types: decision support systems [32, 33], reservation and market-based systems [34, 35], neural networks [36, 37], genetic algorithms [20, 39], fuzzy logic [40–42], reinforcement learning [43, 44]. Their latest and influential work is reviewed as follows.

(1) Decision support systems: Shi and Li [32] described a novel framework of decision support system prototype integrated data base, expert system and knowledge creation. By improving the decisional context to increase the efficiency of assessments and scheduling, this system aims to aid environmental planners, urban designer or transport administrators. A decision support system was also proposed by Shumin et al. [33], which is applied for the urban traffic emergency scheduling based on the expert system. In the emergency incident, their system can formulate strategies of traffic control quickly and effectively. Most systems can offer the effective scheduling, but they are depended on judgments and historical traffic data, and hard to consist with the change of traffic scenarios.
(2) Reservation and market-based systems: Bazzan et al. [34] proposed an agent-based method, which considers some variants for the demand processing and the routing. Vasirani and Ossowski [35] designed a competitive computational market, where driver agents trade the use of the capacity inside the intersections with intersection manager agents. They show how the market dynamics influence the drivers' behavior, leading to a more efficient use of the urban road traffic

system, in terms of lower average travel times and less congestion
(3) Neural networks: Park et al. [36] investigated various methodologies for the traffic information prediction, and presented a speed prediction algorithm. Their algorithm is trained with the historical traffic data, and can predict the vehicle speed profile with the current traffic information. Based on back propagation neural network method, Huang et al. [37] presented a scheduling algorithm, which is alterable in the phase cycle. After considering the lengths of each phase motorcade, their algorithm determines how much time the current phase of the green light to extend. But the urban road networks in control just have very simple structures. And those methods of neural networks suffer degraded performances when traffic volumes change.
(4) Genetic algorithms (GA): Sanchez-Medina et al. [20] developed and tested a new model based on the genetic algorithm for the optimization task. And cellular-automata-based simulators are used to evaluate every possible solution. In the work of Teo et al. [39], the genetic algorithm took the current queue length as its input, and it outputted the optimized green time for the intersection. The result of the genetic algorithm is improved by introducing the incoming traffic flow during red time of each phase. However, due to the delayed convergence behavior of the genetic algorithm, the cost of simulation increases seriously for a large size of network.
(5) Fuzzy logic: Karakuzu and Demirci [40] developed fuzzy logic based traffic junction light simulator system for the smart traffic junction light controller. Mehan and Sharma [41] described a fuzzy logic signal controller for a four–way intersection suitable for mixed traffic, including a high proportion of motorcycles. They discuss the traffic control strategy, which dictates the design criteria for the fuzzy logic controller. Hwang and Cho [42] developed a fuzzy logic traffic system that considers the two two-way intersections, which is able to adjust traffic signals in time based on traffic situation. To determine the preferred actions of a traffic signal, a set of rules are used in fuzzy logic signal controllers based on a number of inputs. But it is difficult to generate an effective rule base, especially for complex intersections with a lot of possible phases.
(6) Reinforcement learning: Prashanth and Bhatnagar [43] proposed a reinforcement learning algorithm with function approximation for the traffic signal scheduling. Their algorithm incorporates state-action features and is easily implementable in high-dimensional settings. The work of Desjardins and Chaib-draa [44] designed a multiple-level architecture using reinforcement learning techniques, which is the first step toward a fully functional low-level controller. But most approaches are effective with static traffic distribution, and suffer a performance penalty when parameters describe traffic state fluctuate over time

Although the abundant achievements have been made in the above work. Three main weak points can be summarized as follows.

(1) Many methods just focus on the special urban traffic network with limited urban elements (traffic lights, intersections, roads, and so on). And some methods are even specially designed for a given scenario or a given city.
(2) Because of the limitation of the convergence rate of optimization algorithm, many methods can't achieve satisfactory results in a short computing time. Many latest variants of classical optimization algorithms with the faster convergence speed are not considered.
(3) Some optimization methods use the historical measured data to determine the scheduling of current traffic lights. But the historical data couldn't accurately describe the current traffic condition. And the traffic light scheduling applying historical data suffers three percent performance decay per year [8].

Compared with the six types of methods above, the current study of POS in the traffic light scheduling is just at the early development stage. As an important part of the artificial intelligence field [15], PSO has achieved the substantial improvements. In order to enhance the performance of PSO, many attempts have been made, such as studying particle's neighbor topologies [23, 24], applying comprehensive learning strategies [25], employing cooperative behaviors [26], considering time variant inertia and acceleration coefficients [27]. Compared with the above six types of methods, PSO has many advantages, such as faster computing speed and better global searching ability.

These advantages make it present a tremendous potential to optimize traffic light scheduling. Chen and Xu [16] applied PSO for training a fuzzy logic controller of each intersection by determining the effective green time for each phase of the traffic lights. And a simple network with two basic junctions is used to test their method. Peng et al. [17] presented PSO with isolation niches to control traffic lights, where PSO is tested in a theoretical instance with a restrictive one-way road with two intersections. Their method mainly applies the capacity of isolation niches to maintain the diversity of the swarm. Kachroudi and Bhouri [18] used a predictive model based on a public transport progres-

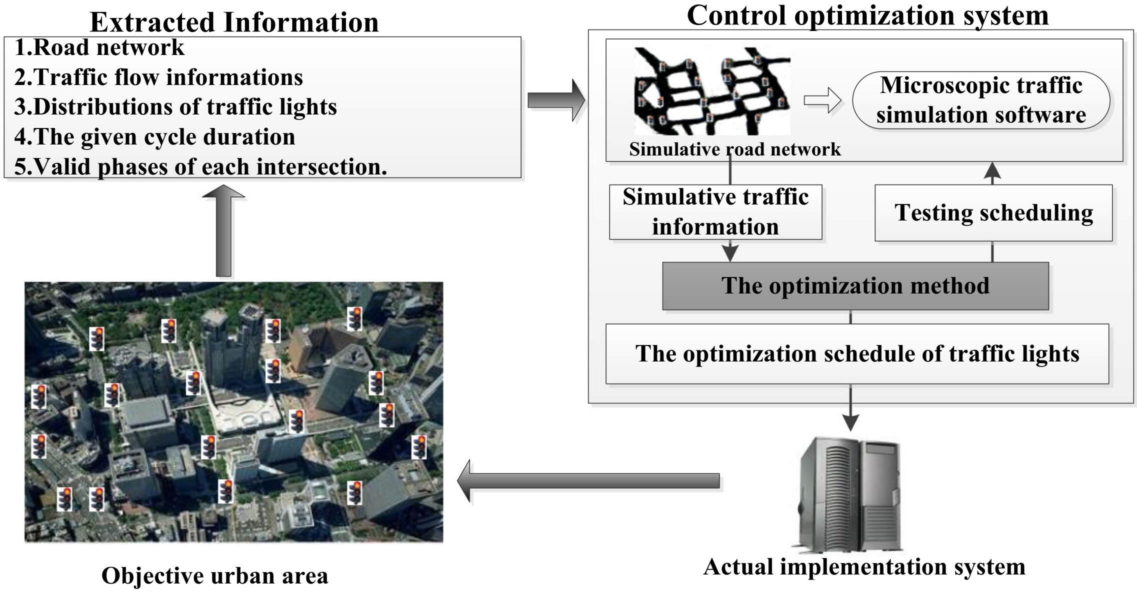


Fig. 1 The framework of the traffic light scheduling

sion model in a multi-objective version of PSO. This work is simulated on a special urban road network of 51 links and 16 intersections. Each intersection is controlled by traffic lights with a same cycle time of 80s. Garcia-Nieto et al. [19] applied PSO to optimize the scheduling of traffic lights in the two different urban networks by simulations in the microscopic traffic simulator. And the sequence of valid phases remains unchanged, which needs be decided correctly by operators in the beginning of optimization. The original PSO algorithm in their method can't provide the enough search capability for an optimal scheduling in the limited time.

In summary, the existing work based on PSO is just designed to achieve the repetitive and monotonous timing scheduling in a given time period. The work [16–18] can only be applied to a theoretical road network. And they can't achieve the flexible optimization scheduling in the actual urban road network. Meanwhile, the existing work [16–19] only focuses on the original PSO method, and many latest and high-efficiency PSO variants still can't find their way into the traffic light optimization. Their monotonous schedules of traffic lights can't solve different special traffic problems flexibly.

To optimize the traffic light scheduling in a large urban area efficiently, the IOCA-PSO method is proposed in this paper, which is a dynamic and real-time traffic optimization method based on PSO and CA mechanism. The discrete and plain rules of the CA mechanism in ICM make it easy to add various rules to control the complex traffic lights. Different rules can be designed to solve different given traffic problems. Meanwhile, the CA mechanism in OCM is integrated into PSO to make the IOPSO algorithm, which makes the performance of the IOCA-PSO method better than the other state-of-the-art methods.

3 Problem description

As the urban traffic lights increase sharply, the traffic light scheduling goes through from the local control in a single intersection to the arterial coordination in a series of intersections, finally focuses on the area scheduling which

Fig. 2 The phase sequence control and the timing control optimization for an intersection

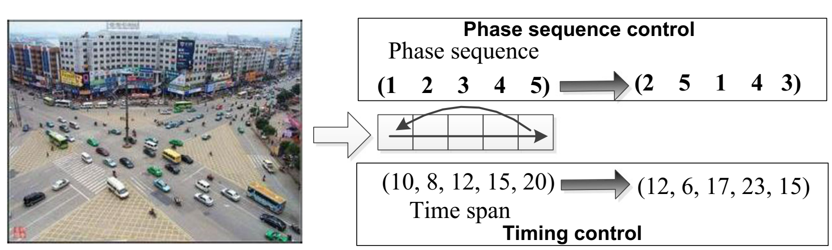
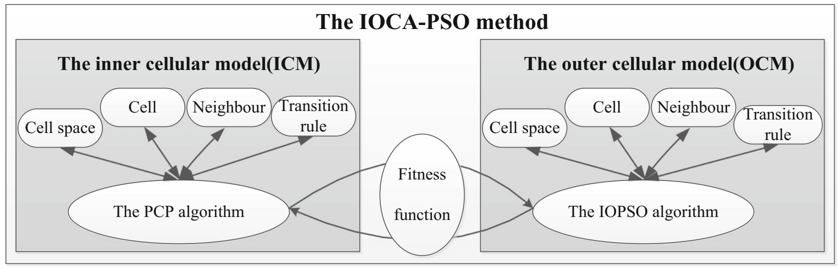


Fig. 3 The framework of the IOCA-POS method



involves hundreds of intersections. The framework of the traffic light optimization scheduling in a large urban area is shown in Fig. 1.

The optimization scheduling of traffic lights can be defined by the pair ($\mathrm{U_f}, \mathrm{J_C}$). $\mathrm{J_C}$ is the objective function. And $\mathrm{U_f}$ is a known scheduling set. This pair is detailed as follows.

(1) The objective function $\mathrm{J_C}$ is an evaluation methodology for different scheduling settings. It is defined below.

$$\mathrm{J_C} : X_{[t_0,t]} + W_{t_0,t} \to R$$

Through the transition of $\mathrm{J_C}$, a ordered pair $\left(\mathrm{X}_{[\mathrm{t_0,t}]}, \mathrm{W}_{[\mathrm{t_0,t}]}\right)$ is mapped to a value in the set of real numbers R. $\mathrm{X}_{[\mathrm{t_0,t}]}$ are the set of the traffic volumes of different roads from $\mathrm{t_0}$ to t. $\mathrm{W}_{[\mathrm{t_0,t}]}$ are the set of the traffic states from $\mathrm{t_0}$ to t. And the mapping value in the set R denotes the comprehensive evaluation of the optimization scheduling from $\mathrm{t_0}$ to t.

(2) $\mathrm{U_f}$ is a known total scheduling set, which contains all possible schedules of traffic lights that make vehicles and pedestrians go through intersections.

The optimization method ties to find the optimal scheduling set U^* in $\mathrm{U_f}$ according to $\mathrm{J_C}$. Assuming $\mathrm{u}(\cdot)$, $\mathrm{u}^*_\mathrm{C}(\cdot)$ are specific scheduling sets ($\mathrm{u}(\cdot), \mathrm{u}^*_\mathrm{C}(\cdot) \in \mathrm{U_f}$), J^*_C is the accessible optimal value ($\mathrm{J}^*_\mathrm{C} \in \mathrm{R}$). And the optimal scheduling set can be understood as the accessible scheduling with the maximal or minimum value J^*_C. It is described as follows.

When $\underset{\mathrm{u}(\cdot)\in\mathrm{U_f}}{\mathrm{Max}} \mathrm{J_C}\left(\mathrm{u}(\cdot)\right) = \mathrm{J}^*_\mathrm{C}$ or $\underset{\mathrm{u}(\cdot)\in\mathrm{U_f}}{\mathrm{Min}} \mathrm{J_C}\left(\mathrm{u}(\cdot)\right) = \mathrm{J}^*_\mathrm{C}$, the optimal scheduling set is determined.

$$U^* = \left\{u^*_C(\cdot) | u^*_C(\cdot) \in U_{f,}\ J_C\left(u^*_C(\cdot)\right) = J^*_C\right\}$$

In the practical optimization, the pair ($\mathrm{U_f}, \mathrm{J_C}$) is what we focus on. $\mathrm{J_C}$ can consider many optimization goals, such as the average delay, the number of stops, and the queuing length of vehicle. But it is not possible to optimize all goals at the same time. $\mathrm{J_C}$ usually consider one or two goals to select the optimal scheduling based on different aims.

All traffic lights at a same intersection have to be necessarily synchronized for the security, and they carry out a series of common phases. A phase of an intersection includes a combination of all traffic lights' color states at the intersection, and the corresponding time span. When an intersection is at a phase, all traffic lights at the intersection carry out their corresponding color states, and keeps their color states unchangeable in the corresponding time span. To avoid the vehicle collisions and accidents, the phases are always kept valid [22], which must follow traffic rules and road constructions. To restrict optimizations to only work with feasible states, only valid phases are considered. The phase in the following paper is short for the valid phase. The schedule in $\mathrm{U_f}$ contains the comprehensive scheduling of traffic lights, which includes the timing control, the phase sequence control and the special phase controls for different kinds of traffic problems. It can be described as follows.

(1) All selected phases at an intersection make up a phase sequence. And the intersection will go through the phase sequence periodically. To ensure that the traffic flow in every direction can own a moving chance, each selected phase in the phase sequence must obtain a time span to carry out.
(2) The optimization of the phase sequence control is to find a more effective sequence of all selected phases. And the optimization of the timing control is to find the best time span of each phase in the phase sequence.

Table 1 The process of the IOCA-PSO method

INPUT:

The road network, the traffic flow information, distributions of traffic lights, the given time period, the phases of each intersection

PROCEDURE:

Divided the given time period into a sequence of cycle durations

For each cycle duration in the cycle duration sequence

Step 1: Establish cell, cell space, neighbor, transition rules of ICM and OCM.

Step 2: Implement the PCP algorithm and the IOPSO algorithm in ICM and OCM cooperatively.

Step 3: Determine the minimum cycle process for the current cycle duration.

Step 4: Output the global optimal scheduling of traffic lights U^*.

End for

Table 2 Symbol definitions

Definitions	Description
t	The count of time-step units in simulations, and initializes as 1.
i	The number of an intersection in the objective urban road network, and initializes as 1
j	The number of a phase in a phase sequence, and initializes as 1.
h	The number of a cycle duration in the cycle duration sequence of a given time period, and initializes as 1. $h_{\max}$ denotes the number of the maximum cycle duration.
k	The number of a particle in the particle swarm in OCM.
g	The number of an iteration in the whole simulative iterations, and initializes as 1. $g_{\max}$ denotes the number of the maximum iteration.
H	The amount of vehicles appearing in simulations in a given cycle duration.
M	The amount of intersections in the objective urban road network.
ph_i^j	The phase j at the intersection i
$ph_{\max}$	The maximum length of all phase sequences.
L	The size of whole particles swarm in OCM
$\sum_{e=0}^{H} TT_e$	The total running time of all vehicles appearing in our simulations in a cycle duration.
$\sum_{e=0}^{H} WT_e$	The total waiting time of all vehicles appearing in our simulations in a cycle duration.
$ts_{(i,j)}$	The time span of the phase j at the intersection i.
$\sum_{1}^{j} ts_{(i,j)}$	The accumulation of time spans from the phase 1 to the phase j at the intersection i.
$g_{(i,j)}, r_{(i,j)}$	The number of green traffic lights, and the number of red traffic lights at the phase j of the intersection i.

For example in Fig. 2, after the optimization of the phase sequence control, the phase sequence of number sequence 1, 2, 3, 4, 5 is adjusted into 2, 5, 1, 4, 3. After the optimization of the timing control, their time spans 10, 8, 12, 15, 20 are adjusted into 12, 6, 17, 23, 15. What's more, there are many different given traffic problems that need special phase controls, such as the greenlight roads with no vehicles.

In the actual application, the scheduling process of U_f consists of two basic parts, the minimum cycle process of traffic lights and its corresponding cycle duration. And all traffic lights repeatedly execute the minimum cycle process periodically in its corresponding cycle duration. Most optimization methods practically are devoted to finding the optimal minimum cycle process for a given cycle duration.

Furthermore, the dynamic and real-time characters of the proposed IOCA-PSO method are described below.

(1) The discovery process of the IOCA-PSO method will be completed at the beginning of each cycle duration. By the actual control system, the optimal minimum cycle process for each cycle duration can be applied for its corresponding actual duration. Because of the gradual change character of the urban traffic flow, the current discovery result also can be applied for the delayed future duration, which can win more time for the execution of the IOCA-PSO method.
(2) The objective time period can be divided into a sequence of successive cycle durations flexibly. Then the IOCA-PSO method will try to find the optimal minimum cycle process for each cycle duration by simulations in the traffic simulator. Because the minimum cycle process for each cycle duration in the objective time period is different, so the IOCA-PSO method is dynamic.
(3) When the acquisition interval of actual traffic information is less than the time span of corresponding minimum cycle processes, and the computing power is enough, the IOCA-PSO method can achieve a real-time scheduling. The time span of each cycle duration is equal to the time span of its corresponding minimum cycle processes. And each minimum cycle process just executes once without repetitions in its corresponding cycle duration.

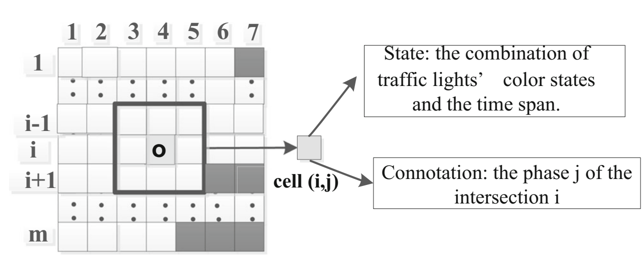


Fig. 4 The state and connotation of cell (i, j)

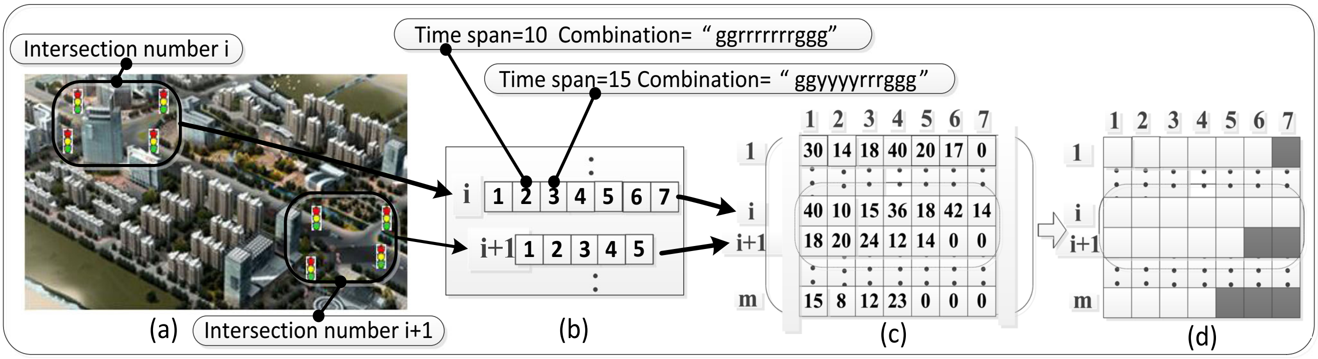


Fig. 5 **a** The real urban network. **b** Phase sequences of different intersections. **c** The $M \times ph_{\max}$ time-span matrix when $ph_{\max} = 7$. **d** The cell space in ICM, and the gray cells denote virtual cells

4 The IOCA-PSO method

In this section, the IOCA-PSO method is detailed, which can achieve a dynamic and real-time optimization scheduling of traffic lights in an extensive urban area, and achieve the global optimal scheduling of traffic lights in the objective urban area. The framework of the IOCA-PSO method is shown in Fig. 3. The IOCA-PSO method consists of ICM, OCM, and the fitness function. (1) The real traffic scheduling has a great deal of problems. ICM is established on its the CA mechanism (cell, cell space, neighbor, transition rule). So that flexible and efficient transition rules of CA can be designed in ICM to solve different traffic problems. ICM make the IOCA-PSO method achieve a comprehensive scheduling. The PCP algorithm in ICM can ensure that the transition rules carry out smoothly. (2) However, it is still hard to artificially determine the color of a huge number of traffic lights at different time. So OCM is proposed to automatically determine the corresponding color state of each traffic light at different time. OCM makes the IOCA-PSO method achieve optimized scheduling. The IOPSO algorithm is the main body of OCM. Based on the CA mechanism (cell, cell space, neighbor, transition rule) in OCM, the IOPSO algorithm can optimize the scheduling of traffic lights more efficiently than other PSO algorithms. (3) In the end, how to verify whether a scheduling by a method is better or not? The fitness function is proposed to evaluate a given scheduling. The content of this section is arranged as follows.

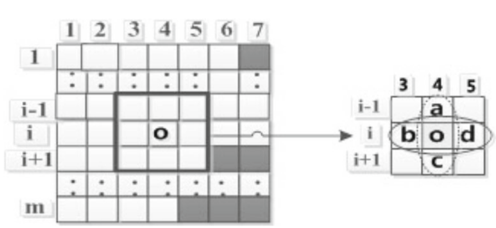


Fig. 6 The vertical and horizontal neighbors of the cell *o*

(1) In Section 4.1, some symbols and definitions are defined to illustrate the IOCA-PSO method, which consists of ICM, the fitness function, and OCM.
(2) Section 4.2 elaborates ICM, which achieves the globally sophisticated scheduling by basic transition rules and affiliated transition rules. In Section 4.4, OCM is devoted to finding the optimal timing control based on the control scheduling conserved in ICM. In ICM and OCM, their corresponding cell, cell space, neighbor, transition rule build the fundamental elements of the CA mechanism. For solving the practical traffic optimization problems, the PCP algorithm and the IOPSO algorithm are designed in ICM and OCM, respectively. ICM is detailed from the cell (in Section 4.2.1), cell space (in Section 4.2.2), neighbor (in Section 4.2.3), transition rule (in Section 4.2.4) and the PCP algorithm (in Section 4.2.5). OCM is detailed from the cell (in Section 4.4.1), cell space (in Section 4.4.2), neighbor (in Section 4.4.3), transition rule (in Section 4.4.4), and the IOPSO algorithm (in Section 4.4.5).
(3) In Section 4.3, the fitness function is detailed, which is the link between ICM and OCM. The fitness function calculates the fitness values of the scheduling setting in ICM. And OCM applies the calculated fitness values to conduct and adjust their optimization of the timing control based on the control scheduling conserved in ICM.

In terms of the pair (U_f, J_C), ICM and OCM offer a coordinated effort to find U^* in U_f, and the fitness function plays the role of J_C. The process of the IOCA-PSO method is described in Table 1.

4.1 Symbols and definitions

Some symbols are defined in Table 2 .

4.2 ICM

The connection of traffic lights in an urban road network is so complex that it is hard to control all traffic lights intuitively and globally. With the help of transition rules in CA, ICM provides a perfect way to control hundreds of traffic lights and develop the solutions for many given problems. ICM is detailed from cell, cell space, neighbor, transition rule, and the PCP algorithm in this subsection. The cell, cell space, neighbor, transition rule make for the CA mechanism in ICM, which conserves the scheduling of traffic lights. The PCP method carries out the conserved scheduling.

4.2.1 Cell

Each cell represents a phase of an intersection. The state of a cell has two basic elements corresponding to its phase, which are a valid combination of traffic lights' color states, and its corresponding time span. A cell in the cell space (detailed in Section 4.2.2) is denoted by its coordinate. For example in Fig. 4, cell (i, j) denotes the phase j of the intersection i. And The Fig. 4 shows the connotation and state of cell (i, j).

4.2.2 Cell space

Each intersection in the objective urban road network is assigned with a number. And the adjacent intersections on a main road are tied to be assigned with the continuous numbers, which is helpful to study and solve the urban trunk road coordination scheduling [28, 29].

All successive phases for one intersection make up a phase sequence. For example, the sequences of the adjacent intersections i and $i+1$ in Fig. 5a are showed in Fig. 5b. The intersection i contains seven phases, and the time spans of the seven phases are 40, 10, 15, 36, 18, 42 and 14s. Twelve

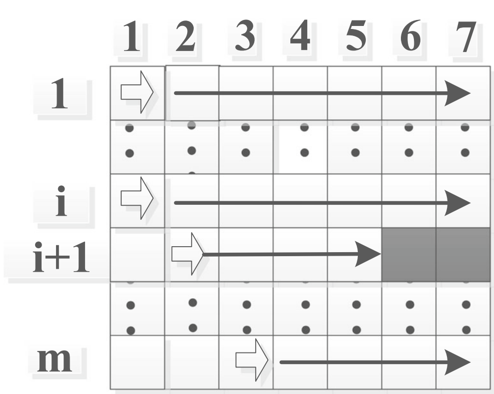


Fig. 7 The sites of some arrows at $t = 25$

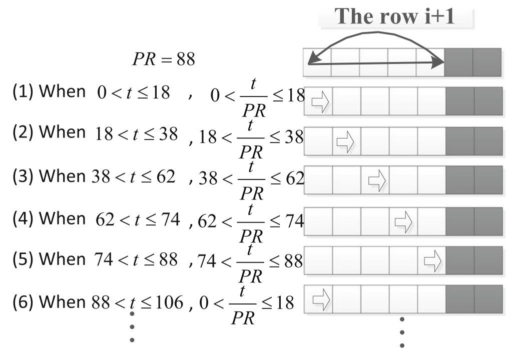


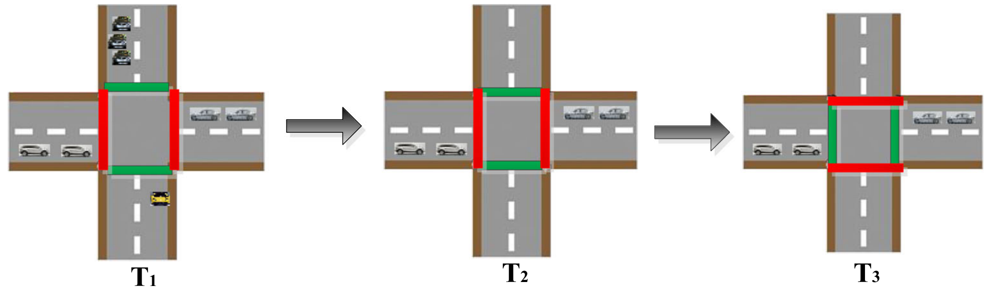
Fig. 8 The moving process of the arrow in the row $i + 1$ (Fig. 5d) at different time steps

traffic lights are located at the intersection i, so the combinations of twelve color states are considered in the seven successive phases. And g, r and y are used to denote that the color state of a traffic light is green, red and yellow, respectively. When the intersection i is at the second phase, the combination of the color states is "ggrrrrrrrggg", which means that the color states of five traffic lights are green (g), and the color states of seven traffic lights are red (r) during the corresponding time span 10s. When the intersection i enters into the new phase, the color states of all traffic lights at the intersection i are adjusted to the corresponding new combination. For instance, the following combination of color states in the third phase is "ggyyyyrrrggg", four red traffic lights will be changed in the yellow color states (y) for next 15s.

The phase sequence lengths of different intersections vary. If the objective urban road network has M intersections, and the maximum length among M corresponding sequences is $ph_{\max}$, the time spans of M independent phase sequences are integrated into a $M \times ph_{\max}$ time-span matrix. Each time span is an element of the time-span matrix. To make the phase sequence of each intersection is $ph_{\max}$, the additional vacant elements are added in the time-span matrix. They are marked by the time span of 0. For example, as is shown in Fig. 5c, the maximum length among M phase sequences is 7. The intersection $i + 1$ (Fig. 5b) has only five phases, which corresponds to the first five time spans in row $i + 1$ of the $M \times ph_{\max}$ time-span matrix (Fig. 5c). As the additional vacant elements, the sixth and seventh elements in row $i + 1$ are marked with the time span of the value 0.

Furthermore, the cell space in ICM is a set of phases of all intersections in the objective urban network, which is cut by the infinite $M \times ph_{\max}$ grids. Each grid creates a cell. The cell in the cell space (Fig. 5d) corresponds to the time span in the time-span matrix (Fig. 5c). And the time span of cell (i, j) can obtained from the element (i, j) in the corresponding time-span matrix. If the corresponding

Fig. 9 Different simulative states at an intersection at T_1, T_2 and T_3 ($T_1 < T_2 < T_3$)



element of a cell in the time-span matrix is an additional vacant element, this cell is called as a virtual cell. The virtual cell is set for maintaining the structural integrity of the cell space. For example in Fig. 5d, the time span of cell $(i, 1)$ is 40s, and cell $(i+1, 6)$, cell $(i+1, 7)$ are virtual cells.

The novel two-dimensional encode of $M \times ph_{\max}$ cell space has three main advantages. (1) This encode is helpful to integrate the CA mechanism into ICM, which is the foundation to add concise rules to achieve the complex traffic scheduling. (2) This encode makes every intersection own independence. The adjustment of each intersection only involves its corresponding row. The scheduling in the cell space can be precise and flexible.

4.2.3 Neighbor

The neighbors of a cell contain two adjacent vertical cells (neighbors) and two adjacent horizontal cells (neighbors). Its two vertical neighbors denote two different phases with the same phase number at different intersections, and its two horizontal neighbors denote its two consecutive phases at the same intersection. Note that the adjacent intersections on one main road are assigned with continuous numbers, so they are mostly adjacent in the cell space vertically.

As an example in Fig. 6, the site of the cell o is (i,4). The cell o has two vertical neighbors a, c and two horizontal neighbors b, d. For cell o, a and c are two different phases of intersections $i-1$ and $i+1$, b and d are its two consecutive phases in the phase sequence of the intersection i. The intersections $i-1$ and $i+1$ are likely to be adjacent with the intersection i in the urban trunk road.

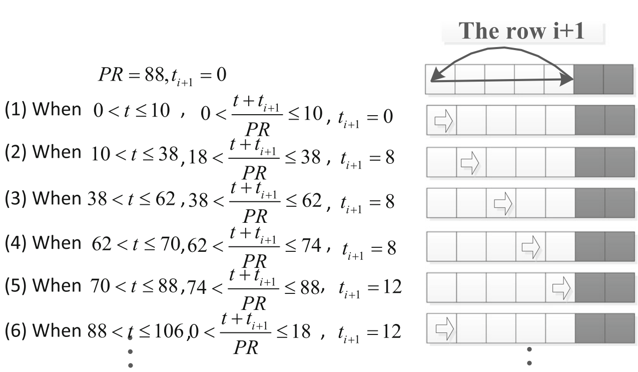


Fig. 10 The moving process of the arrow in the row $i+1$ (Fig. 5d) at different time steps with the affiliated rule $Rule_2$

4.2.4 Transition rule

The CA mechanism of ICM provides an efficient platform to insight the scheduling of traffic lights from the global view. The supreme advantage of the CA mechanism is that concise discrete rules can be defined to achieve the sophisticated control. And the complex and global traffic scheduling in ICM can also be achieved by adding rules. The rules in ICM are classified as the basic transition rules and the affiliated transition rules. They can achieve the timing control, the phase sequence control and the special phase controls for different kinds of traffic problems.

Assuming each intersection is seen as an arrow, the arrow can only move in the cell row whose number is same with its intersection number. The arrow goes through its successive cell in a row periodically. The last cell is followed by the first cell, and this arrow cycle is repeated during the current cycle duration. When an arrow locates in a cell, it means that the intersection denoted by the arrow is at the phase denoted by the cell. Under normal circumstances without the affiliated transition rules, an arrow need stay at a cell up to the corresponding time span of the cell. That is to say, the intersection remains the combination of traffic lights' color states of the current phase unchanged in the time span of the current phase. For example, the arrow denoting the intersection $i+1$ can only move in the row $i+1$ of the cell space. And the current site of this arrow in row $i+1$ is at cell$(i+1, 2)$, which means the intersection $i+1$ is going through its second phase. Figure 7 shows the sites of some arrows at $t = 25$ in Fig. 5d, and different arrows at the same time can locate in different phases.

(1) The basic transition rules are used to determine the phase shifting control of traffic lights, which can also be seen as the right site of each arrow in the cell space at different time steps. PR is introduced to calculate

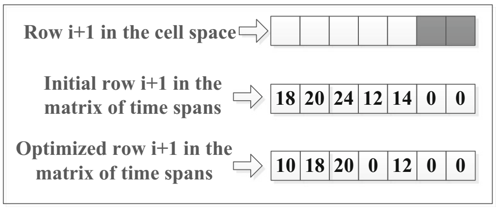


Fig. 11 The change of time spans for the cell row $i + 1$ in the cell space (Fig. 5d)

the accumulation of all phases' time spans at the intersection i, which can be denoted $PR = \sum_{1}^{ph_i} ts_{(i,j)}$.

At each time step t, each arrow in the cell space carries out the basic transition rule $Rule_1$, which can be defined as follows.

Definition $Rule_1$: For an arrow at cell (i, j), if $\frac{t}{PR} - \sum_{1}^{j} ts_{(i,j)} > ts_{(i,j)}$, then $i = i, j = j + 1$. The arrow will enter into the next adjacent cell $(i, j+1)$. Else, $i = i, j = j$, the arrow will still stay at its current cell (i, j).

According to $Rule_1$, the moving process of the arrow in the row $i + 1$ (Fig. 5d) is shown in Fig. 8. Because the time spans of the last two phases are all with the value 0, the intersection goes through first five successive phases. Then it returns to the first phase to start a new moving cycle.

(2) Based on the basic transition rule $Rule_1$, the affiliated transition rules can be made to achieve the timing control and the special phase controls for different kinds of traffic problems.

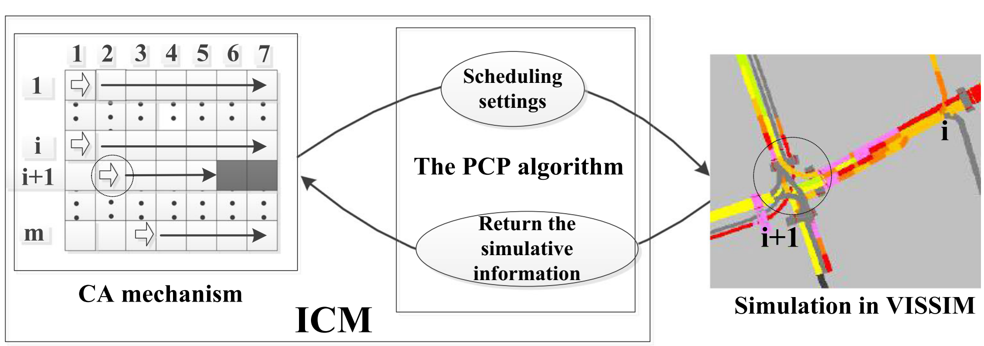
It is a common traffic problem that the roads with green lights have no vehicles to advance. If sensor detections in cities can afford to collect the real-time traffic information, it can be solved in ICM. As an example shows in Fig. 9 at an intersection, T_1 and T_2 ($T_1 < T_2$) are two moments in the common phase j, T_3 is the initial moment of the next phase (the phase j + 1 or the first phase). The east-west and north-south lanes both have waiting vehicles at T_1. After the running phase goes on at T_2, all vehicles on the north-south lanes have leaved, while the vehicles on the east-west lanes with the red lights are still waiting. When this condition is discovered, the current phase j has no effect on improving local traffic flow again. So it is ideal to force the intersection into the next phase at T_3, which make the vehicles on the east-west lanes go through at once. Meanwhile, these operations of the special phase controls are expected not to affect the following cycles of the phases.

For the above condition that the roads with green lights have no vehicles, the solution can be defined by the affiliated transition rule $Rule_2$ For an arrow at cell (i, j), $tb_{(i,j)}$ is the consuming time of the arrow at the current cell. The remaining time of the arrow that need wait at the current cell can be calculated by $ts_{(i,j)} - tb_{(i,j)}$. When the intersection i at the phase j is forced into the next phase (the phase j + 1 or the first phase), $ts_{(i,j)} - tb_{(i,j)}$ is called as the jump time. t_i is the accumulation of all jump time at the intersection i, and initializes as 0. At each time step t, each arrow in the cell space carries out the affiliated transition rule $Rule_2$, which can be defined as follows.

Definition $Rule_2$: If the condition that the roads with green lights have no vehicles is discovered at the phase j of the intersection i at the time step t, then $i = i$, $j = j + 1$, $t_i = t_i + \left(ts_{(i,j)} - tb_{(i,j)}\right)$. And the arrow at the cell (i, j) will enter into the next adjacent cell $(i, j + 1)$ immediately. Else if this condition is not discovered, when $\frac{t+t_i}{PR} - \sum_{1}^{j} ts_{(i,j)} > ts_{(i,j)}$, then $i = i, j = j + 1$. The arrow will enter into the next adjacent cell $(i, j + 1)$. Else, $i = i, j = j$, the arrow will still stay at its current cell (i, j).

Different from the basic transition rule $Rule_1$, the affiliated transition rule $Rule_2$ considers the t_i. The moving process of the arrow in the row $i + 1$ at different time steps is shown in Fig. 10. The condition that the roads with green lights have no vehicles is discovered at $t = 10$ and $t = 70$. Then the intersection terminates the current phase at $t = 10$ and $t = 70$, and is forced into the next phase by adding t_{i+1}.

Fig. 12 The interaction between ICM and the simulator (VISSIM) by the PCP algorithm



Furthermore, the addition of affiliated transition rules can also adjust the existing unreasonable phase sequence. It is an absolute imperative when the optimization of current phase sequence is not satisfying. For the phase j at the intersection i, if the time span of ph_i^j (not including phases of virtual cells) is smaller than the setting threshold value for the time span after the optimization. It can be speculated that the phase ph_i^j has no obvious effect on its corresponding traffic flow. So the phase sequence should be rebuilt to make ph_i^j more effective. The sequence of all phases at the intersection i will be randomly produced again. Then the IOCA-PSO method is implemented to find the optimal scheduling again. For examples in Fig. 5d, the change of time spans of row $i+1$ is shown in Fig. 11. When $ph_{i+1}^4 = 0$ is discovered after the optimization, the phase sequence of row $i+1$ is rebuilt randomly.

For the above condition that needs change the phase sequence, the solution can be defined by the affiliated transition rule $Rule_3$, which can be defined as follows.

Definition $Rule_3$: After the optimization, for each row i in the cell space if there is one phase in the phase sequence whose time span is less than the threshold value , then $ph_i = random(ph_i)$. The phase sequence of the row i will be randomly produced. After the inspections of each row, a new start that finds the optimal scheduling will be executed, which is based on the adjusted phase sequences of the rows.

4.2.5 The PCP algorithm

As is shown in Fig. 12, the PCP algorithm in ICM provides an interaction between the ICM and the urban traffic simulator. It is unrealistic to test the scheduling in ICM by the tentative and repetitive applications in the real urban road network. But the scheduling in ICM can be tested by the urban traffic simulator. VISSIM is a famous simulator, which will be detailed in Section 5.4.3.

Based on the CA mechanism of ICM (cell, cell space, neighbor, transition rule), the PCP algorithm can achieve the complex scheduling of traffic lights in the simulator by executing concise rules. The PCP algorithm performs the scheduling transition rules (the basic transition rules, the affiliated transition rules) for different cycle durations of different time periods. The process of the PCP algorithm is detailed in Table 3.

4.3 Fitness function

In order to evaluate the performances of different optimization methods, the fitness function is proposed. As is shown in Fig. 3, the fitness function links ICM with OCM.

After an iterative simulation in the simulator under a given scheduling, the traffic information will be obtained. The fitness function makes use of the necessary information to compute the corresponding fitness value of the given scheduling. The fitness function is defined in the equation (1). The scheduling conserved in the CA mechanism of ICM for a cycle duration h is denoted as S_{ICM}^h.

$$fitness\left(S_{ICM}^h\right) = \alpha \times \frac{WT}{TT} + \beta \times \frac{Tr}{Tg} + \psi(t)\ (0 < \alpha < 1, 0 < \beta < 1) \tag{1}$$

$$TT = \sum_{e=0}^{H} TT_e \tag{2}$$

$$WT = \sum_{e=0}^{H} WT_e \tag{3}$$

$$Tg = \sum_{i=0}^{M} \sum_{j=0}^{ph_i} ts_{(i,j)} \times g_{(i,j)} \tag{4}$$

$$Tr = \sum_{i=0}^{M} \sum_{j=0}^{ph_i} ts_{(i,j)} \times r_{(i,j)} \tag{5}$$

The main purpose of equation (1) is to minimize the proportion of the waiting time to the running time $\left(\frac{\mathrm{WT}}{\mathrm{TT}}\right)$, and the proportion of the red light time to the green light time $\left(\frac{\mathrm{Tr}}{\mathrm{Tg}}\right)$. A scheduling with the lower fitness value indicates the better simulative performance. The proposed fitness function is detailed as follows:

(1) The vehicles' total trip time includes the total running time (TT) and the total waiting time (WT). TT

Table 3 The process of the PCP algorithm

INPUT:
1. The cell, cell space, neighbor and rule set in ICM.
2. The divided cycle duration sequence of the given time period.
3. Road network, traffic flow information, the distribution of traffic lights.

PROCEDURE:

```
Initialize the settings of VISSIM.
FOR each cycle duration in the cycle duration sequence
  FOR t = 0 to the cut-off time step of the cycle duration.
    Execute the basic transition rules and the corresponding
     affiliated transition rules
    Get the simulative information from VISSIM.
  END FOR
END FOR
```

and WT are formulated by (2) and (3), respectively. When a vehicle's route is decided, if the vehicle always keeps moving to its destination, its ideal running time that arrives at its destination is fixed. The waiting time mainly derives from traffic congestions and red lights. And the excessive waiting time can produce a higher delay to arrive at the final destinations. The waiting time is not essential for vehicles to research their final destinations. So TT is enlarged, and WT is diminished in the fitness function.

(2) Tgdenotes the total green time of all traffic lights, and it can be formulated by (4). Tr denotes the total red time , and it can be formulated by (5). For all intersections, the phases with more green traffic lights are more likely to circulate more vehicles, while the phases with more red traffic lights are more likely to circulate fewer vehicles. The final proportion of states should promote these states with more green traffic lights, and restrain others with more red traffic lights. So Tg is enlarged, and Tr is diminished in the fitness function.

(3) α and β are the weights of the proportions $\frac{\mathrm{WT}}{\mathrm{TT}}$ and $\frac{\mathrm{Tr}}{\mathrm{Tg}}$, respectively. By changing the value of α and β, the fitness function can be adjusted for different aims. The simulative information obtained from the traffic simulator may have deviations from the actual traffic, so the modification value $\psi(t)$ is added to adjust the fitness value.

The standard PSO and variances of PSO are usually applied to find the optimal solutions of mathematical functions theoretically [13, 14]. Without considerations for practical applications, they can simply use the existing mathematical functions as the fitness functions. But in the actual application of traffic scheduling, the meaningful traffic parameters should be considered in the fitness function with no unified standards (Evangelos et al.). To better illustrate our fitness function (SFF), it is compared with the related fitness function (PFF) [19]. The advantages of SFF can be listed as follows.

(1) SFF is effective and concise. PFF just puts the parameters that need decrease on the numerator , and that need increase on the denominator. The relationships among different parameters are disorder in PFF. The V in PFF is dynamic and hard to be measured. When a vehicle is allocated a route, WT and TT in SFF can comprehensively show the driving conditions, including the change of V.

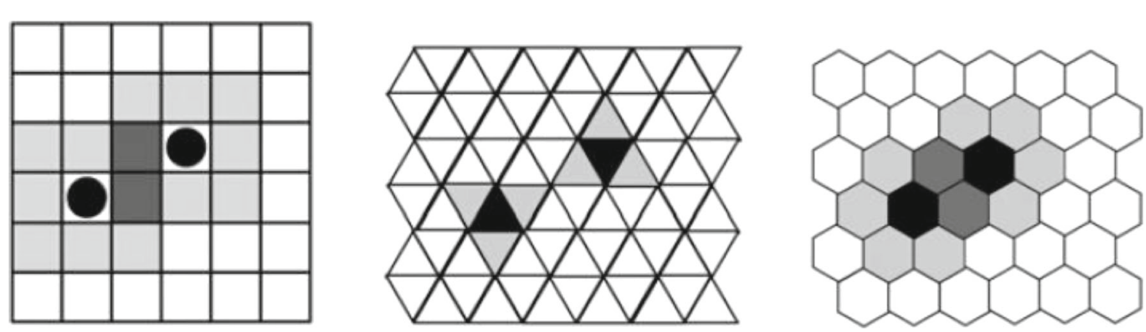
(2) Instead of static fitness function PFF, SFF can be adjusted for different traffic optimization aims. $\frac{\mathrm{Tr}}{\mathrm{Tg}}$ is a typical indicator for the urban traffic condition. And $\frac{\mathrm{WT}}{\mathrm{TT}}$ is associated with drive feelings. SFF can achieve a comprehensive balance at different cases by adjusting α and β.

(3) In SFF, the modification value $\psi(t)$ is introduced to remedy the simulative deviation from the actual traffic information. The fitness function PFF doesn't consider the precision of traffic simulator, and establish on the ideal condition.

(4) The fitness value of SFF is valuable to be compared under different traffic conditions with different vehicle numbers or different simulative time spans. But the vehicle number and the simulative time span are parameters in PFF, PFF is affected seriously by vehicle numbers and the simulative time.

4.4 OCM

OCM is an efficient and fast model to find the optimal timing scheduling for ICM. The time-span matrix conserved in the CA mechanism of ICM is adjusted in OCM for the better timing scheduling. The cell, cell space, neighbor and transition rule constitute the CA mechanism in OCM, which are hybridized with PSO to make the IOPSO algorithm.



(a) Cubic structure (b) Trigonal structure (c) Hexagonal structure

Fig. 13 Three typical lattice structures of CA

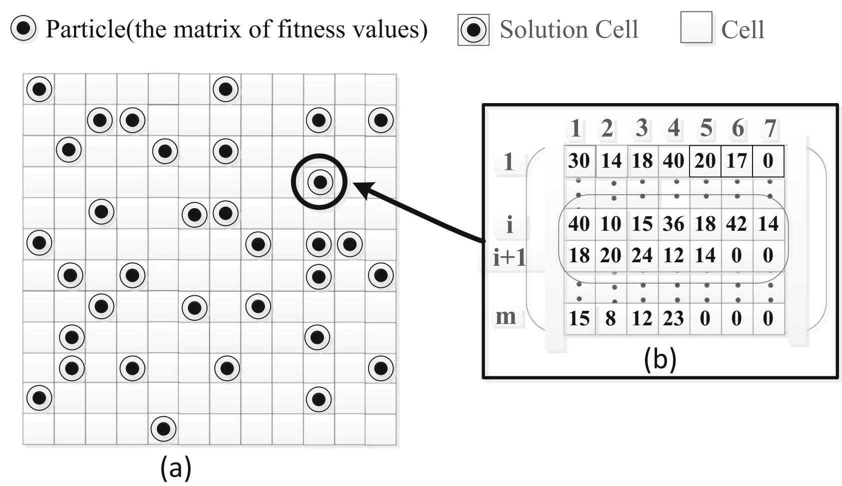


Fig. 14 The particles in the cell space randomly mapped by assigning S^h_{ICM} with different time-span matrixes

4.4.1 Cell

Each cell is seen as a possible position of the search space, which is similar to a position of birds' flying space. Each particle denotes a potential candidate solution, which is similar to a bird. Each cell can contain unlimited number of particles. A cell with one or more particles is called as a solution cell.

The time-span matrix for S^h_{ICM} (a given the scheduling conserved in ICM) is denoted as MS^g_k, and is short for MS. Note that the optimization in OCM is just for the timing control by adjusting the time-span matrix of ICM. The S^h_{ICM} with a given time-span matrix is mapped into the cell space as a particle randomly. As is shown in Fig. 14a, S^h_{ICM} is assigned with different time-span matrixes, which produces different mapping particles in the cell space.

4.4.2 Cell space

A set of cells constitutes the cell space, which is similar to the birds' flying space. Based on different cell structures, the cell space can be defined differently. Three typical lattice structures are shown in Fig. 13. For the briefness and easy comprehension, the cubic lattice structure is used as an example in the paper. As is shown in Fig. 14a, it is assumed that the cell space is cut by the infinite $N \times N$ grids. Every virtual grid is seen as a cell (in Section 4.4.1), which can't be subdivided any more. The size N is determined by the simulative need. An iteration in OCM is accompanied by a complete simulation under a given scheduling setting S^h_{ICM} in ICM.

4.4.3 Neighbor

The neighboring cells of a solution cell (a cell with one or more particles) in the cell space don't simply contain its adjacent cells, which are determined by its neighboring radius. The optimal time-span matrix makes S^h_{ICM} in ICM achieve the smallest fitness value in the simulation. The personal optimal time-span matrix that the particle k has created until the iteration g is stored as MP^g_k. The global optimal time-span matrix that the neighbor particle set of the particle k has ever created until the iteration g is stored as MG^g_k. The distance between the particle k and the farthest particle among all particles is RL. T is the total iterations for the current cycle duration.

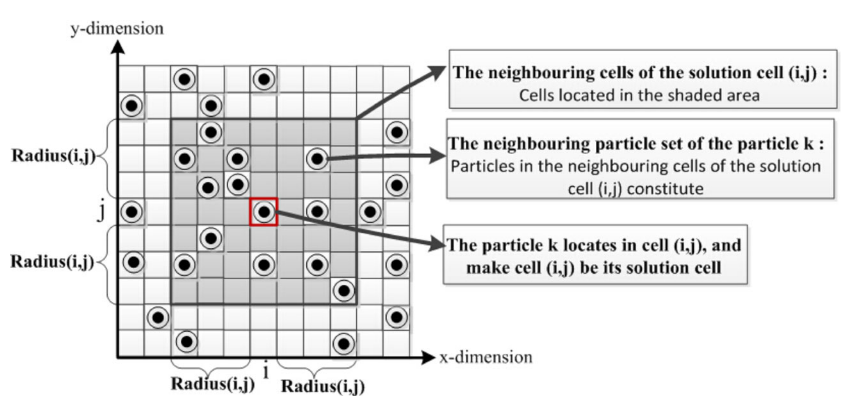


Fig. 15 The neighboring particle set of the particle k for the x-dimension and y-dimension

For the particle k, the neighboring radius of its solution cell (i, j) is denoted as $Radius(i, j)$, which is defined by (6) and (7).

$$\mathrm{Ra} = \begin{cases} \left|\dfrac{fitness\left(MP_k^g\right)}{fitness\left(MG_k^g\right)}\right| \times \frac{t}{T} \times (RL+1), \left(fitness\left(MG_k^g\right)! = 0\right) \\ \dfrac{e^{fitness\left(MP_k^g\right)}}{e^{fitness\left(MG_k^g\right)}} \times \frac{t}{T} \times (RL+1), \left(fitness\left(MG_k^g\right) = 0\right) \end{cases} \tag{6}$$

$$Radius\,(i, j) = \begin{cases} \lfloor Ra \rfloor, & if\ Random(0,1) \le \lambda \\ \lceil Ra \rceil, & otherwise \end{cases} \tag{7}$$

Because the length unit of a cell is 1, the parameter λ determines the probability of performing the ceil or floor function for the intermediate parameter Ra. At the early iterations, the range of $\left|\frac{fitness(MP_k^g)}{fitness(MG_k^g)}\right|$ or $\frac{e^{fitness(MP_k^g)}}{e^{fitness(MG_k^g)}}$ can be small, because the difference of MP_k^g and MG_k^g is relatively large. As the simulative iterations go on, all particles continually converge to the particle with the best fitness value, MP_k^g is adjusted towards to MG_k^g. At finally, the difference continually reduces, and $\left|\frac{fitness(MP_k^g)}{fitness(MG_k^g)}\right|$ or $\frac{e^{fitness(MP_k^g)}}{e^{fitness(MG_k^g)}}$ tends to be the uniform value 1. When all particles converge to a neutral cell after enough iterations, $RL = 0$, $\frac{t}{T} = 1$, and $Radius(i, j) = 1$.

Revolving around cell (i, j), and its neighboring cells can be obtained from function (8). As is shown in Fig. 15, all cells in the shaded quadrate region are the neighboring cells of the solution cell (i, j), which constitutes the neighboring cell set $NS(i, j)$.

$$\begin{aligned} NS(i,j) = \{(a,b) \mid & \min(1, i - Radius(i,j)) \\ & < a < \min(i - Radius(i,j), N), \\ & \min(1, j - Radius(i,j)) \\ & < b < \min(j - Radius(i,j), N)\} \end{aligned} \tag{8}$$

For a particle k located in its solution cell (i, j), its neighboring particle set only contains all particles (include itself) in the neighboring cells of its solution cell (i, j). As is showed in the shaded quadrate area of Fig. 15, the particle k locates in its current solution cell (i, j), so its neighboring particle set are the twelve particles that locate in the neighboring cells of its solution cell (i, j),

4.4.4 Transition rule

To get the lower fitness value, the particle moves inside the cell space to detect its promising solution cells. There are two steps. Firstly, the particle selects the particle with the lowest fitness value in its neighboring particle set. Secondly, the particle will jump into the solution cell of its selected particle. If the selected particle is itself, the particle will not move in the current iteration.

At each iteration g, the transition rule $Rule_4$ in OCM determines the move of each particle in the cell space, which can be defined as follows. SI_k^g is the site of the particle k at the iteration g, which is denoted by the coordination of its solution cell. For each particle k, l_k is the amount of its neighboring particles at the current iteration.

Definition $Rule_4$: Assuming SI_k^{g+1} is $\left(x_k^{g+1}, y_k^{g+1}\right)$, $SI_{k+\delta}^g$ is $\left(x_{k+\delta}^g, y_{k+\delta}^g\right)$, if

$$\begin{aligned} & fitness\left(MS_\phi^g\right) \\ & = min\left(fitness\left(MS_k^g\right), fitness\left(MS_{k+1}^g\right), \ldots, fitness\right. \\ & \quad \left. \times \left(MS_{k+\delta}^g\right), \ldots, fitness\left(MS_{k+l_k}^g\right)\right) \end{aligned}$$

Table 4 The processing procedure of the IOPSO algorithm

Input:
1. The cell, cell space, neighbor, transition rule in OCM.
2. The divided cycle duration sequence of the given time period.
3. The road network, the traffic flow information, the distribution of traffic lights.

Establish ICM of the objective urban network.
Initialize the settings of VISSIM.
For $h = 1$ to the maximum number of the cycle duration in the cycle duration sequence.
    Choose the suitable lattice structure (cubic, trigonal, hexagonal).
    Initialize OPS of each particle in the particle swarm.
  For $g = 1$ to the maximum iteration $g_{\max}$
    For $k = 1$ to the swarm size L
    Determine its neighboring particle set by (6), (7) and (8)
    Carry out the transition rule $Rule_4$ to update SI_k^g.
    Determine MV_k^g and MS_k^g by (9), (10) and (11)
      $MP_k^{g+1} = MP_k^g$, $MG_k^{g+1} = MG_k^g$
      if fitness$\left(MS_k^{g+1}\right)$ <fitness$\left(MP_k^{g+1}\right)$ then
        $MP_k^{g+1} = MS_k^{g+1}$
      End if
      if fitness$\left(MP_k^{g+1}\right)$ <fitness$\left(MG_k^{g+1}\right)$ then
        $MG_k^{g+t} = MP_k^{g+1}$
      End if
     End
  End
  Select and output the optimal time-span matrix MG_k^{g+1} for S_{ICM}^h.
End

where $\phi = k + \delta$, and $fitness\left(MS^g_\phi\right) = fitness\left(MS^g_{k+\delta}\right)$, then $\mathrm{SI}^{g+1}_{\mathrm{k}} = \mathrm{SI}^{g}_{\mathrm{k}+\delta}$, $x^{g+1}_k = x^g_{k+\delta}$, $y^{g+1}_k = y^g_{k+\delta}$. The particle k will jump from the cell site SI^g_k to the cell site $SI^g_{k+\delta}$ at the end of the iteration g, and its updated solution cell site at the iteration $g + 1$ is $SI^g_{k+\delta}$.

4.4.5 The IOPSO algorithm

The IOPSO algorithm is an improved and modified version of the PSO in OCM. The IOPSO algorithm integrates PSO with the above CA mechanism in OCM. And the combination of PSO and CA mechanism is inspired by Shi et al. [14]. Their work has proved the final converge of the cellular particle swarm optimization algorithm theoretically, when the inertia weight of the velocity changes with iterations. And their algorithm performs better than other variants of PSO on many benchmark test functions. More detail of the computational study can be consulted in the work of Shi et al. [14]. But they can't find the way to handle discrete variables to solve real-world optimization problems, their work just seeks for an optimal solution for mathematical functions without practical application values.

Benefit from the CA mechanism of the proposed IOCA-PSO method setting the traffic scheduling background, the IOPSO algorithm is proposed. It is a novel combination of CA and PSO. And it firstly finds the approach to optimize the global timing control of traffic lights. The timing control is of great importance to the traffic light scheduling.

The IOPSO algorithm continually adjusts the time-span matrixes of each mapped particle during the process of continuous iterations. And the IOPSO algorithm is devoted to finding the optimal time-span matrix for a given scheduling S^h_{ICM} in ICM. Along with the continuous iterations, the time-span matrix and the changing velocity are adjusted, which is denoted by (9) and (11).

θ_1 and θ_2 are the weights allowed by the actual need. θ_t changes linearly in the process of optimization iterations. At the beginning, θ_t is introduced with a high value $\theta_{\max}$, which will decrease until reaching the lowest value $\theta_{\min}$. The dynamic θ_t promotes the exploration and exploitation capability. ζ and η are two uniform random values in [0,1]. MS^g_k is the time-span matrix of the particle k at the iteration g. MV^g_k is the changing velocity of the current time-span matrix MS^g_k.

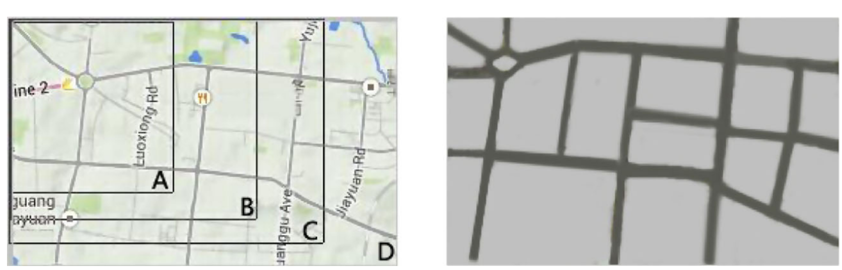
$$MV^{g+1}_k = \theta_t MV^g_k + \theta_1 \xi\left(MP^g_k - MS^g_k\right) + \theta_2 \eta\left(MG^g_k - MS^g_k\right), \quad (9)$$

$$\theta_t = \theta_{max} - \frac{(\theta_{max} - \theta_{min}) \times \mathrm{g}}{g_{max}} \quad (10)$$

$$MS^{g+1}_k = MS^g_k + MV^{g+1}_k \quad (11)$$

The process of the IOPSO algorithm is detailed in Table 4. Differ from most variants of PSO, the IOPSO algorithm has three predominant features, which are detailed as follows.

(1) Instead of owning static neighbors, each particle can select its neighbors in the cell space independently and dynamically.
(2) The move of a particle in the cell space is not just for entering a better solution cell with a smaller fitness value. Its fundamental purpose is to find an optimal time-span matrix for the scheduling conserved in the CA mechanism of ICM for a given cycle duration. So the IOPSO algorithm not only changes cell site SI^g_k of each particle, but also adjusts MS of each particle by adjusting corresponding updating velocity MV. The CA mechanism for the IOPSO algorithm helps the particle to make a wise jump, which can effectively prevent the particle to fall into the local optimization too early. It enhances the diversity of the swarm, and makes the IOCA-PSO method have more potential to seek for the optimal solution in the search space.
(3) Most variants of PSO only remain at the theory stage, but the IOPSO algorithm of the IOCA-PSO method is of high practical value. Along with the change of particle states in iterations, the IOPSO algorithm tries to make each particle in a better solution cell with the lower fitness value. At the final iteration, the IOPSO algorithm determines the global optimal timing control of all traffic lights in the objective area for each given cycle duration.
(4) Within the short-lived computing time, the IOPSO algorithm can't always ensure the final convergence



(a) (b)

Fig. 16 The actual urban area **a** from Google maps map is mapped to the road network **b** in VISSIM

Table 5 The description of some typical time periods in the Wuhan case

Time period description	Time span	Average vehicle number	Average vehicle velocity
The congestion period	8:00–8:30	1237	19 km/h
The half free flow period	10:00–10:30	863	38 km/h
The free flow period	21:00–21:30	531	55 km/h

of all particles, which is different from the hypothetical condition [14]. So that the limit of the IOCA-PSO method is that it just tries to achieve the traffic light scheduling approaching the best scheduling.

5 Experiments and discussions

In this section, sound experimental evidences are provided, which show the characters of the IOCA-PSO method. This section is arranged as follows.

(1) Section 5.1 introduces the Wuhan case and the corresponding traffic information. Our method is tested in the real urban case with the actual traffic data.
(2) The settings of basic and dynamic parameters are detailed in Section 5.2. And the effect of the dynamic parameters α and β in the fitness function is studied in experiments, which shows the flexible use of the proposed fitness function for different aims.
(3) In Section 5.3, the IOCA-PSO method is exhaustively compared with three typical methods, the GA method, the PSO method and the RANDOM method. In Section 5.3.1, the computing complexity analysis of three methods is detailed, which is important for an optimization method to carry out successfully in the limited time. Based on parameter settings in Section 5.2, the optimization results by three methods for different time periods in the Wuhan case are detailed in Section 5.3.2, which show the good performance of the IOCA-PSO method. Furthermore, the effects of three methods on different vehicle numbers and different intersection numbers are compared in Sections 5.3.3 and 5.3.4, respectively. The IOCA-PSO has advantage over three comparison methods with the same traffic settings.
(4) To better study the IOCA-PSO method, its behavior with different vehicle and intersection numbers is observed in Section 5.4.

5.1 Case and data

To develop a practical method for real urban cases, a tested urban case is selected randomly as an example in this section. As is shown in Fig. 16a, an area covering approximately 2.5 km^2 in Wuhan is used as the objective urban area. The area centers the optical valley, which is a prosperous business district with the dramatic change of traffic flow. In this area, the numbers of traffic lights at an intersection change from 6 to 20, and most intersections connect with three and four streets. The main avenues crossing this urban area are: Luoxiong Road, Jiayuan Road, Guanshan avenue, Guanggu avenue, Luoyu Road, Xiongchu avenue, the SBI street, and Gaoxin avenue (http://www.ditu.google.cn/). Four different ranges of the objective urban area (area A, area B, area C, area D) in the Wuhan case are denoted in Fig. 16a. Area D denotes the whole objective urban area.

The urban traffic is simulated by the simulator VISSIM. Its simulation includes vehicles, directions, streets, obstacles, traffic lights, routes, speed [4, 5]. Different from most less-complex models using the constant speeds and deterministic car following logic, VISSIM uses the psycho-physical drive behavior model developed by WIEDEMANN [21]. Through the COM interface, VISSIM can serve as a simulative toolbox under the user-defined signal control logic. And the information in the simulative process can be returned to the user.

By extracting actual information from the digital map, a lot of information is considered into the simulator VISSIM: traffic element locations, the number of lanes, roads, intersections, etc. Each experiment includes two parts: the traffic

Table 6 The setting of some basic parameters in the Wuhan case

Parameter	Value
Simulation time	600 s
The amount of intersections M	17
Maximal iteration time $g_{\max}$	300
The maximum cycle duration $h_{\max}$	3
Swarm size L	400
The maximum length $ph_{\max}$	12
$M \times ph_{\max}$	17×12
The size $N \times N$ in OCM	50×50
Amount of traffic lights	195
Time span scope	4–60 s

simulation and the method implementation. All traffic simulations are carried out in VISSIM to obtain the traffic information, which is used to evaluate the scheduling of traffic lights. The optimization methods are implemented in the VC++6.0 environment [30, 31], which control the signal logic of traffic lights in VISSIM through the COM interface. All experiments are controlled by a distributed task scheduler, and each task is executed as one independent run [6, 7].

Based on the main road networks in Fig. 16a, the simulated view in VISSIM is showed in Fig. 16b. According to the traffic management bureau in Wuhan (http://www.whjg.gov.cn/), the traffic information in the main roads is obtained every 10 minutes, which is collected by sensor detections. Three typical time periods in April 1, 2014 are selected in the following experiments to verify the IOCA-PSO method. They are detailed in Table 5.

5.2 Parameter setting

5.2.1 Basic parameter setting

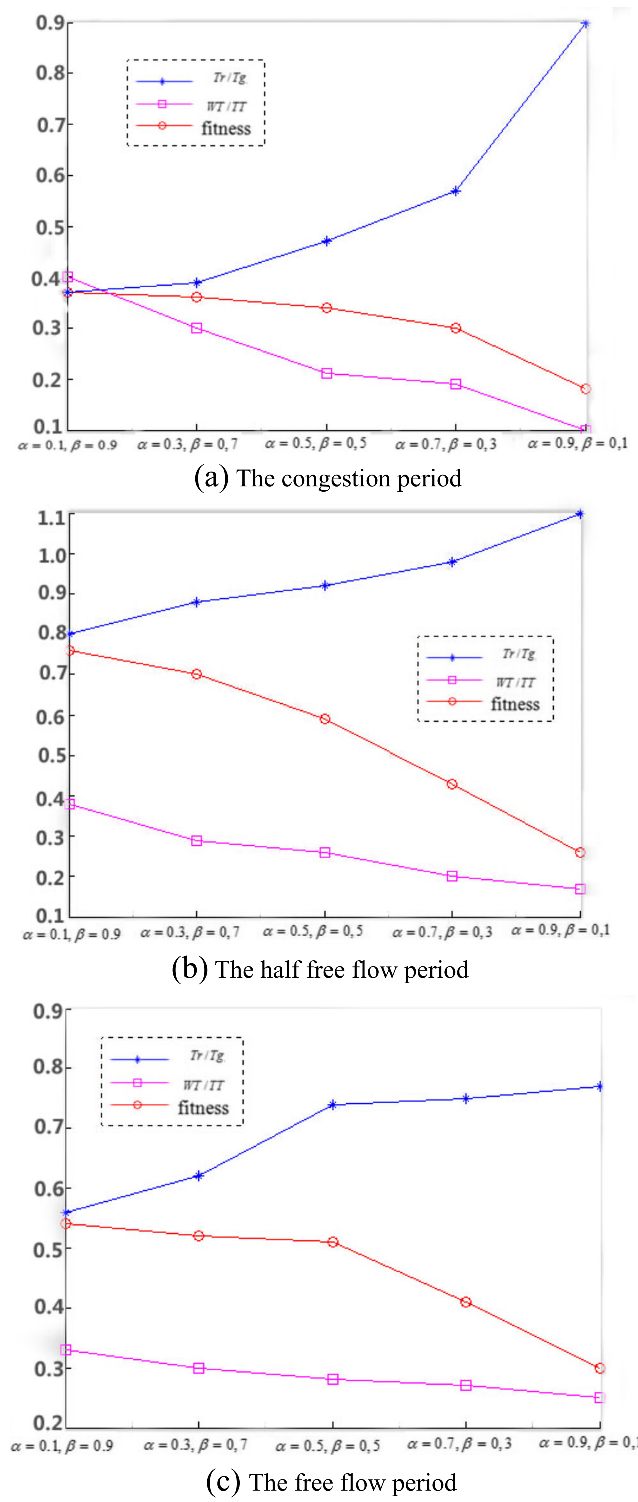
Before the executions of our method for the Wuhan case, some basic parameters were set in Table 6. They are detailed as follows.

(1) The Wuhan case has 17 intersections and 195 traffic lights. Based on the actual circumstance, the phase array of each intersection are determined. The maximum length $ph_{\max}$ is 12. The additional vacant element in the time-span matrix is marked by time the span of 0. And the time spans of other elements in the 17×12 time-span matrix are produced randomly among 4 and 60s (a common time span set).

(2) For three time periods (the congestion period, the half free flow period and the free flow period), each time period will be divided into a cycle duration sequence of three cycle durations. Because the data of the actual traffic information in each road is obtained every 10 minutes, the time span of each cycle duration is set as 10 minutes. For example, the congestion period from 8:00 AM to 8:30 AM is divided into a cycle duration sequence of three cycle durations (8:00–8:10, 8:10–8:20, and 8:20–8:30). The three cycle durations are called as the first, second, third cycle duration of the congestion period in turn.

Table 7 The setting of dynamic parameters in the Wuhan case

Local coefficient θ_1	1.5
Social coefficient θ_2	1.5
Minimum coefficient $\theta_{\max}$	0.2
Maximal coefficient $\theta_{\min}$	0.8
α	0.5
β	0.5
λ	0.5
$\psi(t)$	0



(a) The congestion period

(b) The half free flow period

(c) The free flow period

Fig. 17 The average final fitness value, $\frac{WT}{TT}$, $\frac{Tr}{Tg}$ of the three time periods with different α and β at the Wuhan case

(3) At the beginning of each cycle duration, the number of vehicles on each road is initialized in the VISSIM, which is based on the actual measured data. The initial velocity of each vehicle is randomly produced among the speed limitation of corresponding roads. Each vehicle performs its own route from its origin to its destination. And their routes were generated randomly from a set of reasonable paths at the beginning.

(4) The corresponding simulation time for a cycle duration in VISSIM is 600s (corresponding to the 10 minutes of each cycle duration), which is a deadline for all vehicles to stop their movement. In VISSIM, the moving speed of a vehicle is less than the maximum speeding limitation of its located road. Because of the actual data measured every 10 minutes, it is assumed that the total vehicle number in the simulated view does not change in the corresponding cycle duration, and each vehicle circulates their routes. In the permit of computer computing power and frequent measured dates, each time period can be divided into unequal and shorter cycle durations so that more excise optimization will be obtained.

(5) The optimization operation of the IOCA-PSO method for the above three time periods can be concluded as the following two steps. Firstly, the IOCA-PSO method is applied successively for the three cycle durations in the congestion period. The swarm size in OCM is 400, and each particle in the swarm will go through 300 iterations. Hence, 120,000 solution evaluations are generated per cycle duration to find the best solution of each cycle duration. Secondly, the optimization method is carried out in the same way for the half free flow period and the free flow period in turn.

5.2.2 Dynamic parameter setting

As recommended by the study about the convergence in the way of integrating CA and PSO [14], θ_1, θ_2 θ_3, $\theta_{\max}$ and $\theta_{\min}$ are selected and determined in the IOCA-PSO method. The values of these dynamic parameter settings are shown in Table 7.

The IOCA-PSO method devotes to offering the real-time traffic light scheduling. Within the short-lived computing time, the maximal iteration can't be larger enough to ensure the possible final convergence of the particles in OCM, which is different from the hypothetical condition [14]. So it must be noted that the best timing control with the smallest fitness value is selected from the historic and current solutions of 400 particles, when the iterations are terminated. Based on the scheduling settings in ICM, the fitness value of the best scheduling is selected as the final fitness value to evaluate the performance of an optimization method.

$\lambda = 0.5$ means that $Radius(i, j)$ is ceiled or floored with same probability. To simplify the fitness function, $\psi(t) = 0$ is assumed. α and β are the weights of the proportions $\frac{WT}{TT}$ and $\frac{Tr}{Tg}$, respectively. $\frac{WT}{TT}$ and $\frac{Tr}{Tg}$ are not detached absolutely, which exist a mutual promotion.

As showed in Fig. 17, the average final fitness value of the free flow period is lower than that of the half free flow period, the congestion period relatively. Because the IOCA-PSO method is applied in the free flow period with a better traffic condition. The vehicle number of the free flow period is lower than that of the other two time periods.

And the fitness function can not only evaluate the optimization of a comprehensive scheduling , but also conduct the optimization of the timing control dynamically. By adjust α and β, the fitness function can be used for different optimization aims. The Fig. 17 shows the average final fitness values of the three time periods with different α and β.

Table 8 Computation complexities of different methods

Methods	Procedure				
	Initialize OPS_k^g	Fitness Value Evaluation	Identify MP_k^g and MG_k^g	Update OPS_k^g	Scheduling Implementation
IOCA-PSO (Cubic)	$O(L \times M \times ph_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max} \times M \times ph_{\max})$
IOCA-PSO (Trigonal)	$O(L \times M \times ph_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max} \times M \times ph_{\max})$
IOCA-PSO (Hexagonal)	$O(L \times M \times ph_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max} \times M \times ph_{\max})$
PSO	$O(L \times M \times ph_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max} \times M \times ph_{\max})$
GA	$O(L \times M \times ph_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max})$	$O(L \times g_{\max} \times M \times ph_{\max})$
RANDOM	$O(M \times ph_{\max})$	$O(g_{\max})$	$O(g_{\max})$	$O(g_{\max})$	$O(g_{\max} \times M \times ph_{\max})$

The average final fitness of a time period is the mean value of the fitness values of the corresponding three cycle durations. When α increases, $\frac{WT}{TT}$ decreases. When β increases, $\frac{Tr}{Tg}$ decreases. Meanwhile, the fluctuation of $\frac{WT}{TT}$ is smaller, so when α is higher, corresponding fitness value is higher. And the same results are obtained in the three different time periods.

When α increases to a higher value, $\frac{WT}{TT}$ mainly affects the final value of the fitness function. In the iterations to optimize (decrease) the fitness value, $\frac{WT}{TT}$ will be emphasized to decrease. And β and $\frac{Tr}{Tg}$ is in the same way. To keep the balance between $\frac{WT}{TT}$ and $\frac{Tr}{Tg}$, $\alpha = \beta = 0.5$ is set in below experiments.

5.3 Comparison experiment

The PSO method [19], the GA method [20] and the RANDOM method [18, 19] are introduced as three comparison methods against the proposed IOCA-PSO method. The PSO method and the GA method are the latest works of PSO and GA related to traffic light scheduling, respectively. And the RANDOM method is a typical optimization method. The below experiments are explained as follows.

(1) Because the original aim of the PSO method, the GA method and the RANDOM method is only to optimize the timing control, the IOCA-PSO method in the following experiments only considers basic rules (1) and (2) to achieve a better comparison. The cycle duration sequence and the phase sequence in different time periods are determined by the actual traffic condition in advance.

(2) The fitness function, the swarm size and the running time of three comparison methods are same with the IOCA-PSO method, and their other parameters are set in accordance of their original settings [18, 19].

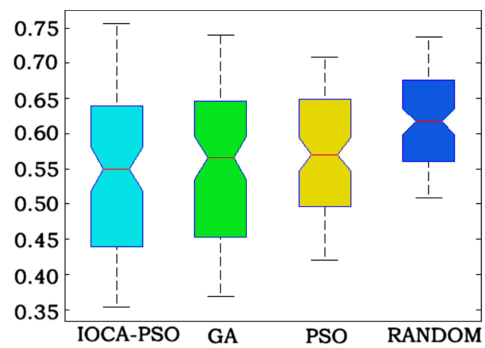
5.3.1 Computing complexity analysis

In the real-time application, the computing time is important. Because the proposed method must find the optimized scheduling in the short-lived computing time, which can ensure that the scheduling is valid for the current cycle duration. The value of the valid time for the current cycle duration is decided by the actual need.

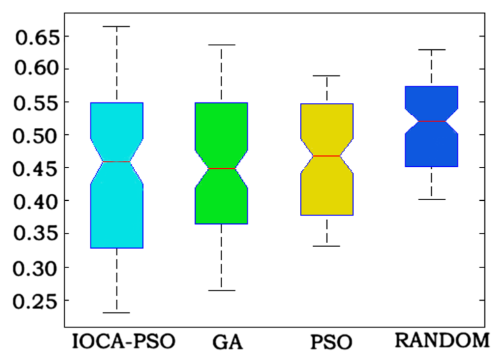
Considering the running environment, programming languages and coding styles, the computing time is not reliable to measure. So the computing complexity analysis is used to evaluate the efficiency of the optimization methods. The runtime in $\circ$ notation is evaluated by counting critical programming statements in iterations. The computing complexities of different methods are showed qualitatively in Table 8. It is summarized as follows.

(1) The IOCA-POS method includes three typical lattice structures, cubic structure, trigonal structure, hexagonal structure. The three typical structures have the same computational complexity. Because of briefness and easy comprehension, the cubic structure is selected in experiments.

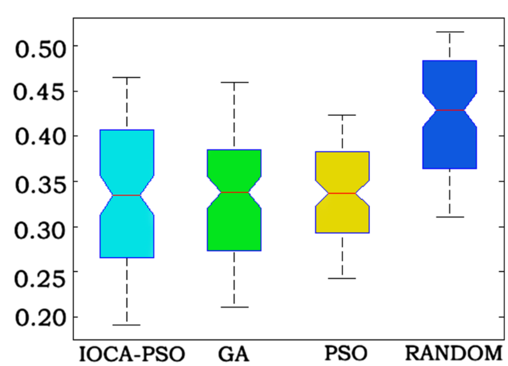
(2) The computational complexity of the RANDOM method is lower than other two methods. Because a swarm is considered the IOCA-PSO method, the PSO method and the GA method, they need evaluate and update the particle in the swarm in each iteration.



(a) The congestion period



(b) The half free flow period



(c) The free flow period

Fig. 18 Boxplots of the appearing fitness values for four methods in three time periods in the Wuhan case

The IOCA-PSO method, the PSO method and the GA method have the same computational complexity.

(3) The computational complexity is mainly affected by the particle swarm size (L), the maximum iteration ($g_{\max}$), the number of intersections (M). A larger swarm size, more iteration or more intersections require more computational efforts. The scheduling implementation causes cost the main computing power.

5.3.2 Comparison results in the three time periods

Figure 18 shows the boxplots of the fitness values for different methods in three time periods, which includes all fitness values appearing in the corresponding time periods. The findings are listed as follows.

(1) Among the four methods, the IOCA-PSO method achieves best performance in limited iterations. In three different time periods, the IOCA-PSO method has the largest fluctuation range and finds the smallest fitness value.

(2) As Table 8 has shown, the IOCA-PSO method, the GA method and the PSO method have the same computational complexity. But the IOCA-PSO is more efficient to find the smallest fitness value, which can schedule the traffic lights of the objective urban area best. The IOCA-PSO method achieves a better balance of the local exploitation and the global exploration, which derives from a powerful CA-based mechanism that guides the site change of particles in OCM.

5.3.3 Comparison results with different numbers of vehicles

To observe the performance of the three optimization methods with different numbers of vehicles, we assume the whole urban area in the Wuhan case with the change of vehicles in simulations. The final fitness values for three methods with different numbers of vehicles are showed in Fig. 19. The findings are listed as follows.

(1) When the vehicle number is less than or equal to 600, the final fitness values of four methods have no distinct difference. Because when the vehicle number is small, almost all vehicles can travel freely without the optimization methods, and the optimization methods can't achieve further improvements on the traffic flow.

(2) When the vehicle number exceeds 600 and less than 2300, the final fitness values of four methods will increase. And this is because the increasing vehicles result in local traffic jams, WT increases and TT decreases. Meanwhile, when the vehicle number increases to cause local traffic jams and is insufficient for global traffic jam, the IOCA-PSO method has obvious advantages to achieve lower final fitness values than three comparison methods.

(3) When global jams happen, the final fitness value tends to infinity. The critical vehicle numbers that lead to global jams by the IOCA-PSO method, the RANDOM method, the PSO method, the GA method are 3000, 2300, 2500 and 2600, respectively. The critical vehicle number that leads to global jams by the IOCA-PSO method is higher than that by three comparison methods.

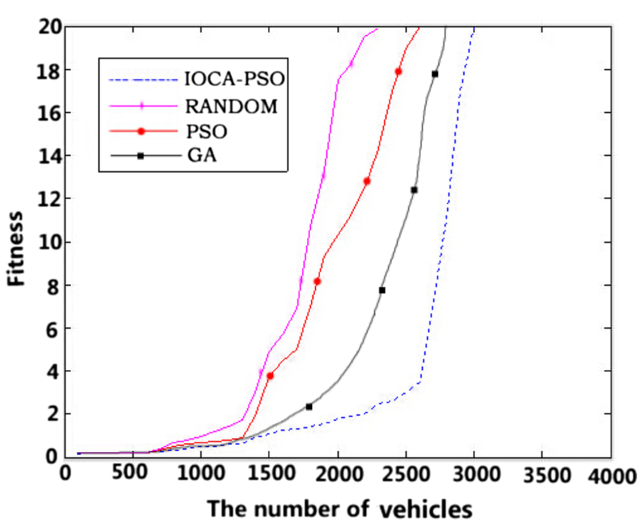


Fig. 19 The final fitness values for four methods in Wuhan case with different numbers of vehicles

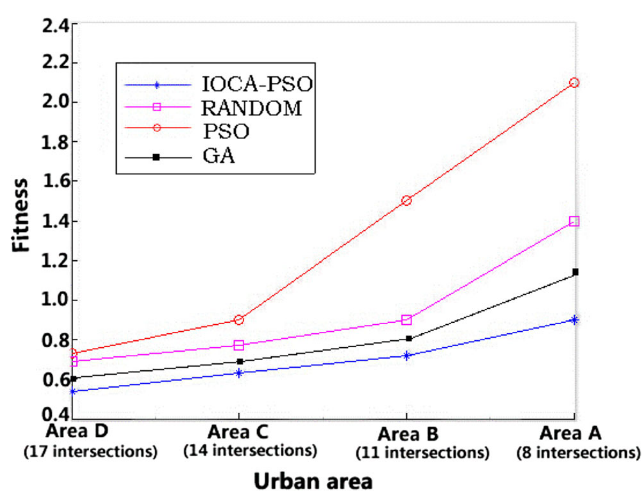


Fig. 20 The final fitness values for four methods in the Wuhan case with different numbers of intersections

5.3.4 Comparison results with different numbers of intersections

To observe the performance of the optimization methods with different numbers of intersections, different ranges of the objective urban area with different numbers of intersections are considered in experiments. As shown in Fig. 16a, the areas of A, B, C and D have 8, 11, 14, and 17 intersections, respectively. In the experiments, all comparison methods should be used to optimize and control the traffic lights in the corresponding range (area A, area B, area C or area D). Other traffic lights not in the corresponding range would not consider in the experiments. The calculation of the final fitness values still considers the whole objective urban area (area D). From the Fig. 20, the findings are listed as follows.

(1) The four optimization methods all have positive effects on the objective urban area. With the control range of intersections increased, the final fitness values of four methods all diminishes. The more traffic lights are in control, the better global scheduling with the smaller fitness value can be discovered by the optimization methods.
(2) The performance of the IOCA-PSO method is better than three comparison methods with different intersections. The final fitness values of the IOCA-PSO method are lower than three comparison methods.

5.4 The performance of the IOCA-PSO method with different numbers of vehicles and intersections

To get a deep insight of the performance of the IOCA-PSO method with different numbers of vehicles and intersections, the change of the final fitness values by the IOCA-PSO method are shown in Fig. 21. The findings can be listed as follows.

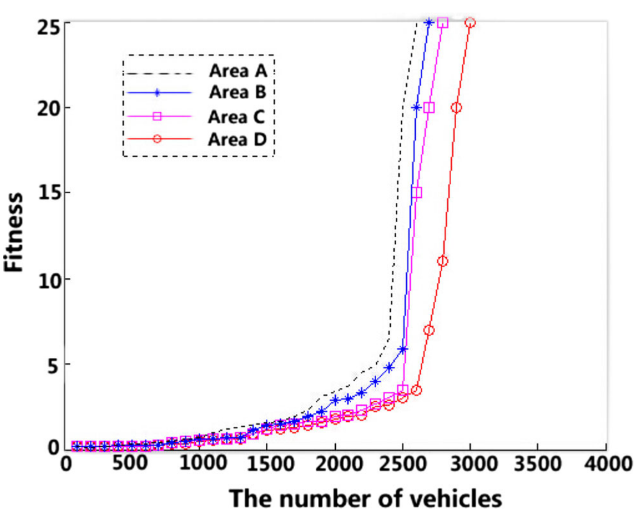


Fig. 21 The final fitness values by the IOCA-PSO method in the Wuhan case with different numbers of vehicles and intersections

(1) The increase of controlling intersections is helpful to find a better scheduling of the objective urban area. When the vehicle number is less than the critical vehicle number that leads to global jams, as the intersections in control increase, the final fitness value will be smaller.
(2) With different numbers of intersections in control, the objective urban area by the IOCA-PSO method has different critical vehicle numbers that lead to global jams. As the number of controlling intersections increases, the critical vehicle number becomes higher. For a special vehicle number less than the critical vehicle number that leads to global jams, the increase of the number of controlling intersections improves the final fitness value.

6 Conclusion

This paper studies the optimization scheduling of traffic lights in the actual large urban road network. The IOCA-PSO method is proposed as a dynamic and real-time optimization method, which consists of ICM, the fitness function, and OCM. The optimization scheduling by the IOCA-PSO method can achieve comprehensive phase scheduling of traffic lights, which includes the timing control, the phase sequence control and the special phase controls for different kinds of traffic problems.

(1) The IOCA-PSO method is compared with the PSO method and the RANDOM method in extensive experiments under different traffic conditions of the Wuhan case. Among the three methods, the IOCA-PSO method achieves the best performance in three different time periods. For each time period, the IOCA-PSO method has the largest fluctuation range and finds the best scheduling with the minimum fitness value in limited iterations.
(2) The IOCA-PSO method achieves a better scheduling than the PSO method and the RANDOM method with different numbers of vehicles. When the vehicle number is small, the final fitness values of three methods have no distinct difference. However, when the vehicle number increases to cause local traffic jams and is less than the critical vehicle number that leads to the global jams, the IOCA-PSO method has obvious advantages to achieve lower final fitness values than three comparison methods and the critical number of vehicles

that leads to global jams by the IOCA-PSO method is higher than that by three comparison methods.

(3) The IOCA-PSO method achieves a better scheduling than the PSO method, the GA method and the RANDOM method with different numbers of controlling intersections. The more intersections are in control, the better global scheduling with the smaller fitness value can be achieved by the IOCA-PSO method.

(4) When the vehicle number is less than the critical vehicle number that leads to the global jams, the increase of controlling intersections and the decrease of vehicles in the objective urban area are conducive to discover a better scheduling with the smaller fitness value by the IOCA-PSO method.

Despite our proposed method being promising, further research is still needed which include (1) how to consider more available realistic features in the evaluation of scheduling results. (2) how to avoid the deviation between the simulative information and the actual information. And the proposed $\psi(t)$ in the fitness function can only provide empirical settings. (3) how to make the IOCA-PSO method more efficient to achieve the better scheduling in the limited time.

Acknowledgments This work is partially supported by the National Natural Science Foundation, China (No.70901060 and 61471274), Hubei Province Natural Science Foundation (No. 2011CDB461), and Youth Plan Found of Wuhan City (No.201150431101). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

1. Kaur T, Agrawal S (2014) Adaptive traffic lights based on hybrid of neural network and genetic algorithm for reduced traffic congestion. In: Recent Advances in Engineering and Computational Sciences (RAECS). IEEE, pp 1–5
2. Tirachini A, Hensher DA, Rose JM (2014) Multimodal pricing and optimal design of urban public transport: the interplay between traffic congestion and bus crowding. Transp Res B Methodol 61:33–54
3. Wang S, Djahel S, McManis J (2014) A Multi-Agent based vehicles re-routing system for unexpected traffic congestion avoidance. In: IEEE 17th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp 2541–2548
4. Lu L, Yun T, Li L, Su Y, Yao D (2010) A comparison of phase transitions produced by PARAMICS, TransModeler, and Vissim. IEEE Intell Transp Syst Mag 2(3):19–24
5. Asamer J, van Zuylen HJ, Heilmann B (2013) Calibrating car-following parameters for snowy road conditions in the microscopic traffic simulator VISSIM. IET Intell Transp Syst 7(1):114–121
6. Chen K, Powers J, Guo S, Tian F (2014) CRESP: towards optimal resource provisioning for mapreduce computing in public clouds
7. Gordon D (2014) The well-connected processor array. IEEE Trans Comput 63(5):1287–1295
8. Bell MC, Bretherton RD (1986) Ageing of fixed-time traffic signal plans. In: International conference on road traffic control
9. Tian JF, Jia N, Zhu N, Jia B, Yuan ZZ (2014) Brake light cellular automaton model with advanced randomization for traffic breakdown. Transportation Research Part C: Emerging Technologies 44:282–298
10. Hu W, Wang H, Min Z (2014) A storage allocation algorithm for outbound containers based on the outer-inner cellular automaton. Information Sciences
11. Elsayed SM, Sarker RA, Mezura-Montes E (2014) Self-adaptive mix of particle swarm methodologies for constrained optimization. Inform Sci:216–233
12. Subrata R, Zomaya AY (2014) A robust adaptive array beamformer using particle swarm optimization for space-time code division multiple access systems. Inform Sci:174–186
13. Neri F, Mininno E, Iacca G (2013) Compact particle swarm optimization. Inform Sci 239:96–121
14. Shi Y, Liu H, Gao L, Zhang G (2011) Cellular particle swarm optimization. Inform Sci 181(20):4460–4493
15. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems (No. 1). Oxford University Press
16. Chen J, Xu L (2006) Road-junction traffic signal timing optimization by an adaptive particle swarm algorithm. In: 9th international conference on control, automation, robotics and vision, 2006. ICARCV'06, pp 1–7
17. Peng L, Wang MH, Du JP, Luo G (2009) Isolation niches particle swarm optimization applied to traffic lights controlling. In: Proceedings of the 48th IEEE conference on decision and control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009, pp 3318–3322
18. Kachroudi S, Bhouri N (2009) A multimodal traffic responsive strategy using particle swarm optimization. In: Control in transportation systems, pp 531–537
19. Garcia-Nieto J, Alba E, Carolina Olivera A (2012) Swarm intelligence for traffic light scheduling: application to real urban areas. Eng Appl Artif Intell 25(2):274–283
20. Sánchez-Medina JJ, Galán-Moreno MJ, Rubio-Royo E (2010) Traffic signal optimization in "La Almozara" district in saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. IEEE Trans Intell Transp Syst 11(1):132–141
21. Wiedemann R (1974) Simulation des strassenverkehrsflusses
22. Leung JY (ed) (2004) Handbook of scheduling: algorithms, models, and performance analysis. CRC Press
23. Suganthan PN (2002) Particle swarm optimiser with neighborhood operator. In: Proceedings of IEEE congress on evolutionary computation, pp 1958–1961
24. Gaing ZL (2003) Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 3:1187–1195
25. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 3:281–295
26. Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evol Compu 8(3):225–239
27. Tripathi PK, Bandyopadhyay S, Pal SK (2007) Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Info Sci 177(22):5033–5049
28. Luo P, Ma Q, Huang H-x (2009) Urban trunk road traffic signal coordinated control based on multi-objective immune algorithm. In: 2009 IEEE international Asia conference on informatics in control, automation and robotics (CAR'09). IEEE, pp 72–76
29. Shen G (2006) Urban traffic trunk two-direction green wave intelligent control strategy and its application
30. Sun Z, Li W, Sha A (2010) In: 2010 sixth international conference on automatic pavement cracks detection system based on visual studio C++ 6.0. natural computation (ICNC), vol 4. IEEE, pp 2016–2019

31. Han X, Xie W, Xiao C (2010) A use of quasi-human algorithm on layout of aircraft. In: 2010 3rd international conference on advanced computer theory and engineering (ICACTE), vol 1. IEEE, pp V1-448-V1-451
32. Shi Y, Li J (2010) Improving the decisional context: new integrated decision support system for urban traffic-related environment assessment and control. In: 2010 International Conference on Mechanic Automation and Control Engineering (MACE). IEEE, pp 1760–1763
33. Shumin S, Zhaosheng Y, Maolei Z (2010) A decision support system of urban traffic emergency control based on expert system. In: 2010 IEEE International Conference on Software Engineering and Service Sciences (ICSESS). IEEE, pp 221–225
34. Bazzan AL, de Brito do Amarante M, Da Costa FB (2012) Management of demand and routing in autonomous personal transportation. J Intell Transp Syst 16(1):1–11
35. Vasirani M, Ossowski S (2011) A computational market for distributed control of urban road traffic systems. IEEE Trans Intell Transp Syst 12(2):313–321
36. Park J, Li D, Murphey YL, Kristinsson J, McGee R, Kuang M, Phillips T (2011) Real time vehicle speed prediction using a neural network traffic model. In: The 2011 International Joint Conference on Neural Networks (IJCNN), pp 2991–2996. IEEE
37. Huang ZJ, Li CG, Zhang ZF (2010) Traffic signal control based on genetic neural network algorithm. In: 2010 International Conference on Intelligent Computing and Integrated Systems (ICISS), pp 31–34. IEEE
38. Sánchez-Medina JJ, Galán-Moreno MJ, Rubio-Royo E (2010) Traffic signal optimization in "La Almozara" district in saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. IEEE Trans Intell Transp Syst 11(1):132–141
39. Teo KTK, Kow WY, Chin YK (2010) Optimization of traffic flow within an urban traffic light intersection with genetic algorithm. In: 2010 2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM), pp 172–177. IEEE
40. Karakuzu C, Demirci O (2010) Fuzzy logic based smart traffic light simulator design and hardware implementation. Appl Soft Comput 10(1):66–73
41. Mehan S, Sharma V (2011) Development of traffic light control system based on fuzzy logic. In: Proceedings of the international conference on advances in computing and artificial intelligence, pp 162–165. ACM
42. Hwang KS, Cho SB (2011) Expert systems with applications
43. Prashanth LA, Bhatnagar S (2011) Reinforcement learning with function approximation for traffic signal control. IEEE Trans Intell Transp Syst 12(2):412–421
44. Desjardins C, Chaib-draa B (2011) Cooperative adaptive cruise control: a reinforcement learning approach. IEEE Trans Intell Transp Syst 12(4):1248–1260

Wenbin Hu was born in 1976. He is currently an associate professor at the School of Computer, Wuhan University, Wuhan city, China. His main research interests are intelligent simulating and optimization, multi-agent system and swarm intelligent algorithm.

Huan Wang was born in 1989. He is currently a master at the School of Computer, Wuhan University, Wuhan city, China. His main research interests are intelligent simulating and optimization.

Liping Yan was born in 1980. She is currently a doctoral student at the School of Computer, Wuhan University, and Wuhan city, China. Her main research interests are intelligent simulating and optimization, multi-agent system.

Bo Du is currently an associate professor with the School of Computer, Wuhan University, and Wuhan, China. His major research interests include complex system, pattern recognition, hyper spectral image processing, and signal processing