Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 814871, 10 pages
http://dx.doi.org/10.1155/2015/814871

Research Article

Study of On-Ramp PI Controller Based on Dural Group QPSO with Different Well Centers Algorithm

Tao Wu,[1] Xi Chen,[2] and Yusong Yan[3]

[1] *School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China*
[2] *School of Computer Science & Technology, Southwest University for Nationalities, Chengdu 610041, China*
[3] *School of Computer Science & Technology, Southwest Jiaotong University, Chengdu 610031, China*

Correspondence should be addressed to Tao Wu; dorawu840720@gmail.com

Received 31 May 2014; Revised 11 November 2014; Accepted 12 November 2014

Academic Editor: Baozhen Yao

A novel quantum-behaved particle swarm optimization (QPSO) algorithm, dual-group QPSO with different well centers (DWC-QPSO) algorithm, is proposed by constructing the master-slave subswarms. The new algorithm was applied in the parameter optimization of on-ramp traffic PI controller combining with nonlinear feedback theory. With the critical information contained in the searching space and results of the basic QPSO algorithm, this algorithm avoids the rapid disappearance of swarm diversity and enhances the global searching ability through collaboration between subswarms. Experiment results on an on-ramp traffic control simulation show that DWC-QPSO can be well applied in the study of on-ramp traffic PI controller and the comparison results illustrate that DWC-QPSO outperforms other evolutionary algorithms with enhancement in both adaptability and stability.

1. Introduction

Interference caused by intersection and convergence between different traffic flows at the traffic bottlenecks such as on-ramp not only seriously reduces the efficiency of freeway but also leads to a reduction of the entire traffic network efficiency. On-ramp control is the most effective way to improve traffic condition of freeway. Therefore, it is necessary to study the traffic flow and develop effective traffic management and control measures on the basis of existing facilities at the on-ramp. With the development of computer technology, computational intelligence provides an effective method which can select the optimal control parameters for ramp traffic control strategy according to different conditions to solve the on-ramp traffic flow control problem. Now, the ant colony optimization (ACO) [1], particle swarm optimization (PSO) [2], artificial neural network (ANN) [3], cellular automata (CA) [4], and genetic algorithms (GA) [5, 6] have been used in on-ramp control.

Although the computational intelligence methods above have made some achievements on on-ramp traffic control, they still have the following limitations: firstly, the convergence rate of PSO, ACO, and GA is so slow as to result in premature convergence and low precision. Secondly, due to the fact that traffic flows have real-time and nonlinear characteristics, the existing models are not fast enough to respond to real-time information, thus having limited applications. To solve these problems, a novel quantum-behaved particle swarm optimization (QPSO) algorithm, the dual-group QPSO with different well centers (DWC-QPSO) algorithm, is proposed in this paper based on quantum-behaved particle swarm optimization (QPSO) algorithm. With the critical information contained in the searching space, DWC-QPSO avoids the rapid disappearance of swarm diversity and enhances the global searching ability through collaboration between subswarms. Then the on-ramp traffic PI controller is designed based on DWC-QPSO algorithm by optimizing the values of parameters K_p and K_i. Experiment results show that DWC-QPSO algorithm can be applicable in the ramp traffic PI controller and the comparison results illustrate that DWC-QPSO outperforms other evolutionary algorithms with enhancement in both adaptability and stability.

This paper is organized as follows. Section 1 gave an introduction of on-ramp control problem; Section 2 introduced the PSO algorithm and QPSO algorithm. Then, a new algorithm, DWC-QPSO algorithm, is proposed in Section 3, and the convergence performance of the novel method is analysed by test functions experiment results; Section 4 explained the on-ramp traffic model and illustrated the computation process of DWC-QPSO algorithm optimizing the parameters of the on-ramp PI controller. In the end, Section 5 focuses on the simulation results analysis and Section 6 is the conclusion of this work.

2. PSO Algorithm and QPSO Algorithm

2.1. PSO Algorithm. Particle swarm optimization (PSO) algorithm is an evolutionary computation method proposed by Kennedy and Eberhart in 1995 [7–9]. It is one pattern of the latest swarm intelligence (SI) optimization algorithms. The basic idea of PSO algorithm derived from the study of bird behavior, through simulating birds' prey behavior to achieve the purpose of problem optimization.

In the PSO algorithm, each potential solution of the problem is abstracted as a particle with no weight and no volume. These particles fly at a certain speed in the D-dimensional search space [10]. Assuming the particle swarm population size is N, the particle i's position and velocity at time step t are, respectively, expressed as follows.

The position of particle i at time step t:

$$X_i(t) = \{x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t)\}. \quad (1)$$

The velocity of particle i at time step t:

$$V_i(t) = \{V_{i1}(t), V_{i2}(t), \ldots, V_{iD}(t)\}. \quad (2)$$

p_{best} is used to describe the particle i's personal best solution and g_{best} is used to describe the best position found by the neighborhood of particle i, at time step t:

$$\begin{aligned} p_{\text{best}} &: P_i(t) = \{p_{i1}(t), p_{i2}(t), \ldots, p_{iD}(t)\}; \\ g_{\text{best}} &: G(t) = \{p_{g1}(t), p_{g2}(t), \ldots, p_{gD}(t)\}. \end{aligned} \quad (3)$$

In the standard PSO model, the updated formula of particles' velocity and position on each dimension is as follows:

$$\begin{aligned} V_{i,j}(t+1) &= wV_{i,j}(t) + c_1 r1_{i,j}(t)\left(P_{i,j}(t) - X_{i,j}(t)\right) \\ &\quad + c_2 r2_{i,j}(t)\left(G_j(t) - X_{i,j}(t)\right), \\ X_{i,j}(t+1) &= X_{i,j}(t) + V_{i,j}(t+1) \end{aligned} \quad (4)$$

for $i = 1, \ldots, N$ and $j = 1, \ldots, D$, where

(i) N is the total number of particles in the swarm,

(ii) D is the dimension of the problem, that is, the number of parameters of the function being optimized,

(iii) w is the inertia weight,

(iv) c_1 and c_2 are acceleration coefficients,

(v) $r1_{i,j}(t), r2_{i,j}(t) \sim U(0,1)$.

As PSO has simple concept, a small number of parameters, and desirable performance, PSO has become a very promising optimization tool and attracted extensive attention and has been successfully applied in many areas, such as function optimization, power systems, data mining, and wireless sensor network. However, some problems of this novel algorithm remain to be solved. One of these problems is that particle swarm cannot converge on the global optimal solution with one hundred percent probability, which has been proven by Bergh [11].

2.2. Quantum-Behaved Particle Swarm Optimization Algorithm. In order to solve this problem, Sun [12] proposed the quantum-behaved particle swarm optimization algorithm in the quantum space. According to the results Clerc and Kennedy [13] analyzed the particles' orbit in the PSO algorithm, and QPSO algorithm establishes a δ potential well to impact particles' convergence at the local attraction points $p_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The position of p_i is calculated by the following equations:

$$p_{i,j}(t) = \frac{c_1 r1_{i,j}(t) P_{i,j}(t) + c_2 r2_{i,j}(t) G_j(t)}{c_1 r1_{i,j}(t) + c_2 r2_{i,j}(t)}, \quad 1 \le j \le D, \quad (5)$$

where $P_{i,j}$ and G_j denote the j-dimensional components of particle i's past best solution (p_{best}) and global best solution (g_{best}), respectively; $r1$ and $r2$ are random factors which are independently distributed random variables; c_1 is the individual cognitive acceleration factor, and c_2 is the group cognitive acceleration factor:

$$\begin{aligned} L_{i,j}(t) &= 2\alpha \cdot \left|C_j(t) - X_{i,j}(t)\right| X_{i,j}(t+1) \\ &= p_{i,j}(t) \pm \alpha \cdot \left|C_j(t) - X_{i,j}(t)\right| \ln\left[\frac{1}{u_{i,j}(t)}\right], \end{aligned} \quad (6)$$

where $u_{i,j}(t) \sim U(0,1)$, $L_{i,j}(t)$ is the length of the δ potential well, which is evaluated by formula (6).

$C(t)$ is defined as the population gravity center which is the average of all the best individual locations. It is calculated as follows:

$$\begin{aligned} C(t) &= \left(C_1(t), C_2(t), \ldots, C_n(t)\right) = \frac{1}{N}\sum_{i=1}^{N} P_i(t) \\ &= \left(\frac{1}{N}\sum_{i=1}^{N} P_{i,1}(t), \frac{1}{N}\sum_{i=1}^{N} P_{i,2}(t), \ldots, \frac{1}{N}\sum_{i=1}^{N} P_{i,N}(t)\right), \end{aligned} \quad (7)$$

where is contraction-expansion coefficient. It is the only additional control parameter in QPSO besides the population size and the number of iterations. α can be set by fixed values or decreases linearly.

3. DWC-QPSO Algorithm

3.1. DWC-QPSO Algorithm. QPSO ensures global convergence; however, this algorithm still has the local optimum problem. The main reason for that is that QPSO algorithm establishes a single δ potential well at the point p_i, and the population of particles gradually approaches to p_i under the gravity function. While this information exchange method is fast, the transmission is unidirectional, leading to the population of particles quickly gather in an increasingly smaller search area and resulting in low population diversity. In order to improve the performance of QPSO algorithm, this paper proposes the DWC-QPSO algorithm, which enhances the QPSO's convergence properties by increasing wait effects between particles to avoid population excessive accumulation.

The specific idea of DWC-QPSO algorithm is as follows: dividing the particle swarm into two separate subgroups after randomizing the individual's position in the solution space. One subgroup establishes a δ potential well at the position of p_i in accordance with the rules of standards QPSO algorithm. The particles of the second subgroup are iteratively searching in the solution space under the influence of the other δ potential well which is established with p_i' as the center. Given the entire population S, the first subgroup is called master subgroup, represented by S_1, and the second subgroup is called secondary subgroup, represented by S_2; thus, $S_1 \cup S_2 = S$. The position of p_i' is calculated as in formula (8).

Evolutionary formula of particles' positions in DWC-QPSO algorithm is as follows:

$$p_{i,j}(t)' = \left[1 - \varphi_{i,j}(t)\right] P_{i,j}(t) + \varphi_{i,j}(t) G_j(t), \quad (8)$$

$$\begin{aligned} X_{i,j}(t+1) &= p_{i,j}(t) \pm \alpha \cdot \left|C_j(t) - X_{i,j}(t)\right| \\ &\quad \times \ln\left[\frac{1}{u_{i,j}(t)}\right] \quad i \in N_{S_1}, \\ X_{i,j}(t+1) &= p_{i,j}(t)' \pm \alpha \cdot \left|C_j(t) - X_{i,j}(t)\right| \\ &\quad \times \ln\left[\frac{1}{u_{i,j}(t)}\right] \quad i \in N_{S_2}. \end{aligned} \quad (9)$$

These two subgroups exchange information by means of the best individual during the searching period in DWC-QPSO algorithm that is comparing the best fitness value of S_1 and S_2 at the end of each iteration. Given that g_{best1} is S_1's global best solution and g_{best2} is S_2's global best solution, we compare the fitness values corresponding to g_{best1} and g_{best2}, respectively. If the latter one is superior than the former one, g_{best2} is assigned to g_{best1}; otherwise g_{best1} remains unchanged. The essence of the above operation is an update of the whole swarm's global best solution g_{best} to help subgroups escape from local optima. By analyzing the displacement formula, the result shows that p_i and p_i' are symmetry in point of the center of p_{best} and g_{best}. In the convergence process, all particles' p_{best} will gradually move closer to the g_{best}. Due to the gravity of δ potential well, p_i of master subgroup and p_i' of secondary subgroup will overlap at the same place in the solution space eventually. Therefore, the convergence of the DWC-QPSO algorithm can be achieved.

3.2. Experiment and Simulation. In order to verify the novel algorithm, the optimization results of DWC-QPSO algorithm are compared with the performance of standard PSO algorithm and QPSO algorithm using 6 standard test functions. Since many practical projects, including single-peak and multipeak problems, choose two single peak functions (Sphere function and Rosenbrock function) and five multipeak functions (Rastrigin function, Griewank function, Ackley function, Schaffer function, and Schwefel function) as the experimental test functions, we will give some information about these test functions, including expressions, the ranges of variables, dimensions, the optimal solutions, and optimal values of test functions as follows:

(1) Sphere function:

$$f_1(X) = \sum_{i=1}^{D} x_i^2, \quad x_i \in [-100, 100], \quad D = 20, \quad \min f_1(0, 0, \ldots, 0) = 0, \quad (10)$$

(2) Rosenbrock function:

$$f_2(X) = \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right], \quad x_i \in [-10, 10], \quad D = 20, \quad \min f_2(1, 1, \ldots, 1) = 0, \quad (11)$$

(3) Rastrigin function:

$$f_3(X) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right), \quad x_i \in [-5.12, 5.12], \quad D = 20, \quad \min f_3(0, 0, \ldots, 0) = 0, \quad (12)$$

(4) Griewank function:

$$f_4(X) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad x_i \in [-600, 600], \quad D = 20, \quad \min f_4(0, 0, \ldots, 0) = 0, \quad (13)$$

(5) Ackley function:

$$\begin{aligned} f_5(X) &= -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) \\ &\quad - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right) + 20 + e, \end{aligned}$$

$$x_i \in [-32.786, 32.786], \quad D = 20, \quad \min f_5(0, 0, \ldots, 0) = 0, \quad (14)$$

TABLE 1: The results of optimization algorithms ($D = 20$, $N = 30$).

Algorithm		f_1	f_2	f_3	f_4	f_5	f_6
PSO	FV	$1.3782E-286$	$1.5113E001$	$1.1550E001$	$4.1815E-002$	$1.6991E-011$	$4.2418E003$
	T (s)	0.8379	0.1753	0.1965	0.1992	0.2147	0.1797
QPSO	FV	$1.3872E-287$	$1.5928E001$	$1.0945E001$	$4.1815E-002$	$1.7319E-011$	$4.2317E003$
	T (s)	0.8297	0.1766	0.2031	0.1984	0.2188	0.1797
DWC-QPSO	FV	**1.0461E − 313**	**1.5672**	**5.9996**	**1.2431E − 002**	**1.8385E − 013**	$1.4243E003$
	T (s)	**0.7563**	**0.1203**	**0.1328**	**0.1216**	**0.1641**	**0.1547**

TABLE 2: The results of optimization algorithms ($D = 20$, $s = 50$).

Algorithm		f_1	f_2	f_3	f_4	f_5	f_6
PSO	FV	$2.1469E-289$	$1.3941E001$	10.1020	$3.7234E-002$	$2.1537E-014$	$4.1132E003$
	T (s)	1.2889	0.2987	0.2966	0.3289	0.3517	0.2655
QPSO	FV	$2.0941E-289$	$1.4351E001$	9.6164	$3.6314E-002$	$2.0428E-014$	$4.0053E003$
	T (s)	1.2984	0.3047	0.2938	0.3391	0.3422	0.2641
DWC-QPSO	FV	**7.1978E − 313**	**9.8871**	**5.9698**	**4.0719E − 006**	**1.3323E − 014**	**9.4751E002**
	T (s)	**1.2906**	**0.2062**	**0.2062**	**0.2625**	**0.2641**	**0.2484**

(6) Schwefel function:

$$\begin{aligned} &f_6(X) = 418.9829D - \sum_{i=1}^{D} x_i \sin\left(x_i^{1/2}\right), \\ &x_i \in [-500, 500], \quad D = 20, \\ &\min f_7(420.98, 420.98, \ldots, 420.98) = 0. \end{aligned} \tag{15}$$

In the experiment simulation, the parameter α linearly decreases from 1 to 0.5. The maximum number of iterations *iterMax* is set to 1000. The population sizes are set to 30 and 50, respectively. The size of the master cluster is N_{S1} and the size of the secondary cluster is N_{S2}, using the average distribution strategy, which is $N_{S1} = N_{S2}$. Each test case runs 30 times independently, then calculate the average results of the test function. Experimental environment is MATLAB 7.9.0, Intel Core (TM) 2, 1.80 GHz, 2.50 GB RAM.

Mean fitness value and CPU time per round of PSO, QPSO, and DWC-QPSO algorithm to optimize the various functions are as shown in Tables 1 and 2.

Tables 1 and 2 illustrate that the DWC-QPSO has the best performance compared with the PSO algorithm and the QPSO algorithm. This is because the DWC-QPSO algorithm has two subgroups, which are master subgroup S_1 and secondary subgroup S_2, respectively. S_1 and S_2 complement each other and coevolve by exchanging the optimal location information constantly.

This pattern of two subgroups searching in solving space at the same time and learning from each other increases the probability of finding the optimal solution and improves the convergence performance of the new algorithm.

Figures 1(a) and 1(b) show the convergence curves of the PSO, QPSO, and DWC-QPSO algorithm solving 20 dimensions of unimodal functions with 30 particles, respectively. Sphere function is relatively simple, which can be used to test the accuracy of optimization algorithms to study the implementation of the algorithm performance. It is learned from Figure 1(a) that DWC-QPSO has the similar performance. Rastrigin's function is difficult to optimize as it has a large number of local optimums. The novel one gets the best results of these three algorithms for solving this function.

Figures 1(c)~1(e) illustrate the convergence curves of the PSO, QPSO, and DWC-QPSO algorithm solving 20 dimensions of multimodal functions with 30 particles, respectively. Rastrigin function is easy to fall into local minima. The pattern of dural subgroups interaction allows DWC-QPSO algorithm to have a better ability to escape from local optima to get the better optimization. From Figures 1(d) to 1(f) are the curves of Griewank function, Ackley function, and Schwefel function. There are correlations between the variables of these three functions. The experiment results show that the DWC-QPSO algorithm has better performance in this type of problem.

In summary, the DWC-QPSO algorithm proposed by this study obtained outstanding global search capability in limited iterative steps because it could get more useful information.

4. The On-Ramp PI Controller Based on DWC-QPSO Algorithm

4.1. On-Ramp Traffic Model. Consider that the on-ramp has one lane, and the main lane of freeway is divided into M sections with length L_i. The traffic model is shown in Figure 2, where $i = 1, 2, \ldots, M$.

The traffic flow model in the ith section of freeway is discrete in both space and time domains. It includes the following variables:

$\sigma_i(k)$: the traffic flow density of ith section during the kth period;

$v_i(k)$: the average vehicle speed in ith section during the kth period;

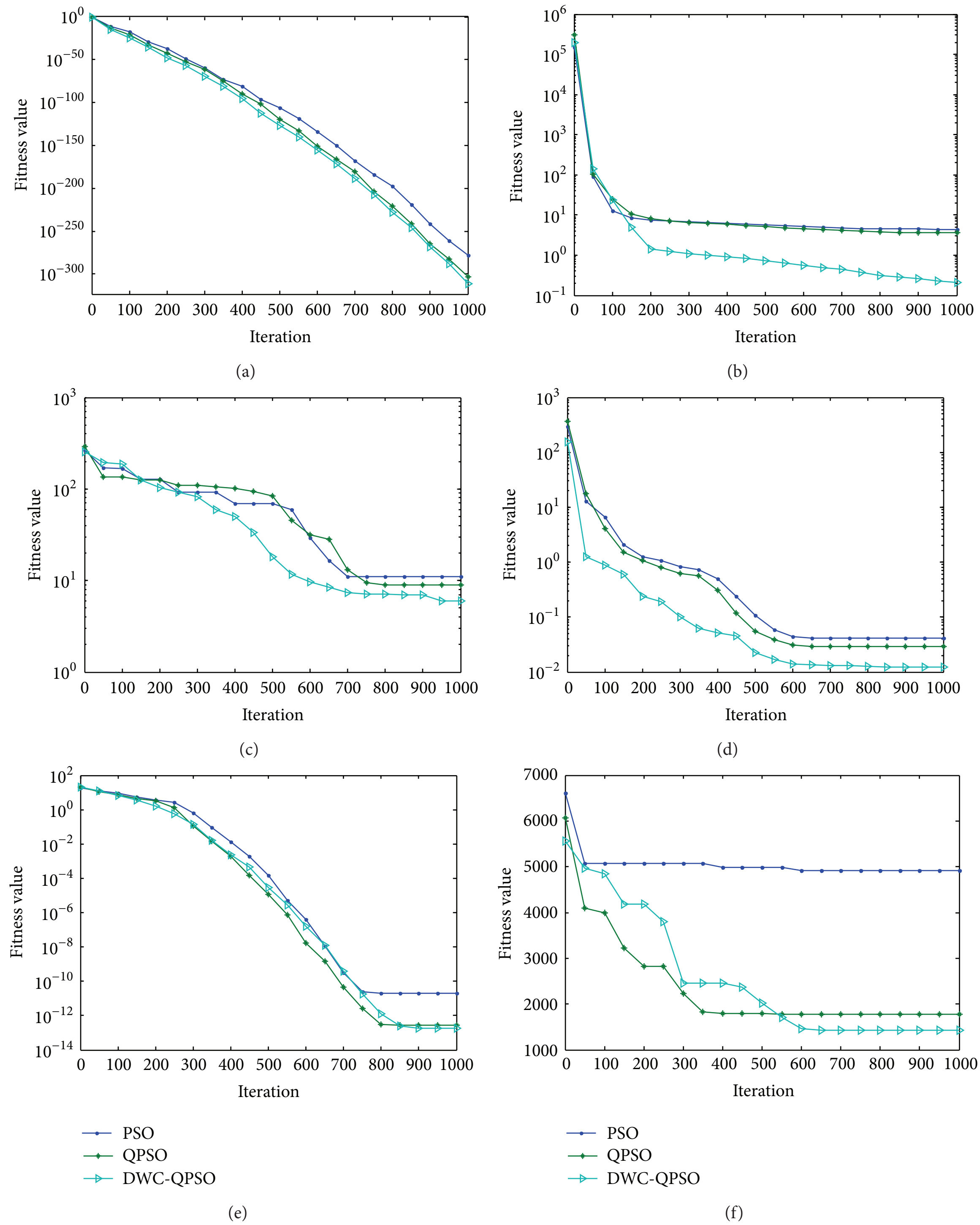

Figure 1: The convergence curves of test functions: (a) f_1, (b) f_2, (c) f_3, (d) f_4, (e) f_5, and (f) f_6.

$q_i(k)$: the increased traffic flow of $i+1$th section from ith segment during the kth period;

$r_i(k)$: the on-ramp metering rate of ith section during the kth period;

$N_i(k)$: the total number of vehicles in ith section during the kth period;

λ: the number of main lanes of freeway;

L_i: the length of the ith section;

T: the discrete time steps.

According to the law of vehicles conservation, the total number of vehicles in ith section during the $(k+1)$th period is as follows:

$$N_i(k+1) = N_i(k) + T\left[\lambda q_{i-1}(k) - \lambda q_i(k) + r_i(k)\right]. \quad (16)$$

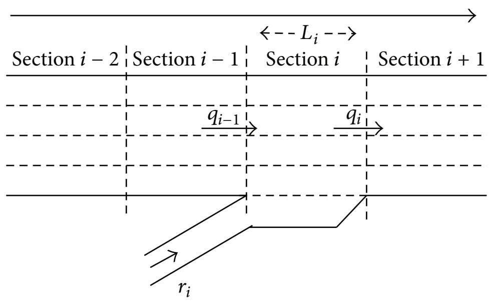

FIGURE 2: On-Ramp traffic model of freeway.

The traffic density of ith section can be calculated through (9) in line with the definition of the traffic density:

$$\sigma_i(k) = \frac{N_i(k)}{\lambda L_i}. \quad (17)$$

Then,

$$\sigma_i(k+1) = \sigma_i(k) + \frac{T}{\lambda L_i}\left[q_{i-1}(k) - q_i(k) + \frac{r_i(k)}{\lambda}\right]. \quad (18)$$

The parabolic flow-density model can be deduced:

$$q_i(k) = v_f \sigma_i(k)\left(1 - \frac{\sigma_i(k)}{\sigma_{\text{jam}}}\right) = v_f\sigma_i(k) - \frac{v_f\sigma_i(k)^2}{\sigma_{\text{jam}}}. \quad (19)$$

After the combination of (16) and (17), the complete traffic flow of the on-ramp of freeway is shown as formula (18):

$$\sigma_i(k+1) = \sigma_i(k) + \frac{T}{\lambda L_i} \times \left[q_{i-1}(k) + \frac{r_i(k)}{\lambda} - v_f\sigma_i(k) - \frac{v_f\sigma_i(k)^2}{\sigma_{\text{jam}}}\right], \quad (20)$$

where v_f is the free-flow speed and σ_{jam} is the jamming density of the freeway. In fact, the flow-density model also creates a critical density σ_c corresponding to the maximum flow rate q_m. $q_i(k)$ increases with the increase of $\sigma_i(k)$ value in $[0, \sigma_c]$ interval, while $q_i(k)$ decreases with the increase of $\sigma_i(k)$ value in $[\sigma_c, \sigma_{\text{jam}}]$ interval. When $\sigma_i(k)$ is equal to σ_{jam}, the $q_i(k)$ is 0; therefore, the traffic jam occurs.

4.2. The On-Ramp PI Controller Based on DWC-QPSO. As a feedback control system, PI controller becomes one of the main methods of engineering controls because of its simple structure, good stability, reliability, and easy adjustment. Performance of PI controller depends on the reasonable values of parameters K_p and K_i. At present, PI controller parameters rely mainly on manual adjustment. However, this traditional approach is not only time consuming but also unable to guarantee the best performance. In this paper, DWC-QPSO algorithm is used to optimize the value of K_p and K_i, which is shown in Figure 3.

PI controller system in Figure 3 includes the following variables:

error value: $e_i(k) = \sigma_{di}(k) - \sigma_i(k)$;

error variation: $\Delta e_i(k) = e_i(k) - e_i(k-1)$;

the output of the PI controller: $\Delta r_i(k) = K_p \Delta e_i(k) + K_i e_i(k)$;

mediation rate of the on-ramp: $r_i(k) = r_i(k-1) + \Delta r_i(k)$;

expected traffic density: $\sigma_{di}(k) = \sigma_c - \varepsilon$, ε is an appropriate small positive number.

This system's objective is to control the mediation rate of the on-ramp $r_i(k)$ and avoid the traffic congestion through maintaining the traffic density of main lane $\sigma_{di}(k)$ at a negative neighborhood of critical density σ_c. Therefore, the square sum of the differences between actual traffic flow density $\sigma_i(k)$ and expected traffic density $\sigma_{di}(k)$ can be used as the system objective function:

$$J = \sum_{k=1}^{n}(e_i(k))^2 = \sum_{k=1}^{n}(\sigma_i(k) - \sigma_{di}(k))^2. \quad (21)$$

Each particle of the swarm corresponds to a pair of values of parameters K_p and K_i when using the DWC-QPSO algorithm optimizing the PI controller. Due to the fact that the square sum of the differences between actual traffic flow density $\sigma_i(k)$ and expected traffic density $\sigma_{di}(k)$ is larger, the corresponding adjustment of on-ramp rate $r_i(k)$ value should be smaller and the DWC-QPSO adaptation algorithm corresponding value should also be smaller; the reciprocal of the sum of squared deviations can be selected as the fitness function of the DWC-QPSO algorithm:

$$\min F = \frac{1}{\sum_{k=1}^{n}(e_i(k))^2} = \frac{1}{\sum_{k=1}^{n}(\sigma_i(k) - \sigma_{di}(k))^2}. \quad (22)$$

Figure 4 illustrates the process of using DWC-QPSO algorithm optimizing the PI controller parameters.

The specific process is as follows.

Step 1 (setting parameters). This includes the acceleration factors c_1 and c_2, the contraction-expansion factor α, the swarm population N, the maximum allowed times of iterations *iterMax* or error accuracy, and the population of subgroups, where N_{S1} and N_{S2} are the population of master subgroup S_1 and secondary subgroup S_2, respectively.

Step 2 (population initialization). For S_1, randomly generate the particles' initial positions $X_{i,j}(0)$ in the solution space, and let $P_{i,j}(0) = X_{i,j}(0)$, where $i = \{1, \ldots, N_{S1}\}$, $j = \{1 \cdots D\}$; for S_2, do the same operation with S_1, generate the particles' random initial positions $X_{i,j}(0)$, and let $P_{i,j}(0) = X_{i,j}(0)$, where $i \in \{N_{S1}+1, \ldots, N\}$, $j = \{1 \cdots D\}$.

Step 3. Calculate the fitness value of all particles in S_1 and S_2 according to (22). The particle location of the smallest fitness value is assigned to the global best solution; that is, $\min(f(X_i))$, $i \in \{1, \ldots, N\} \rightarrow g_{\text{best}}$.

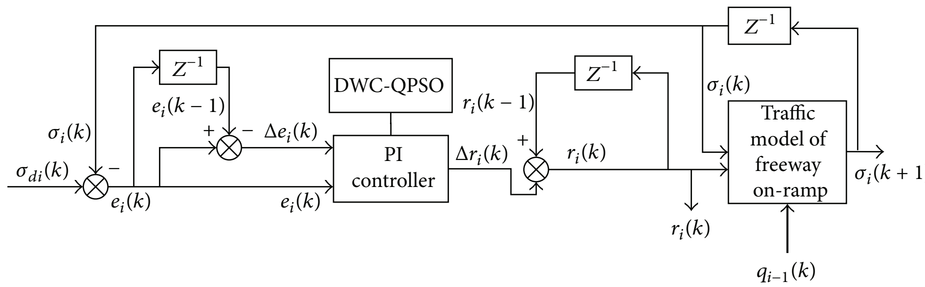

Figure 3: On-ramp PI controller based on DWC-QPSO.

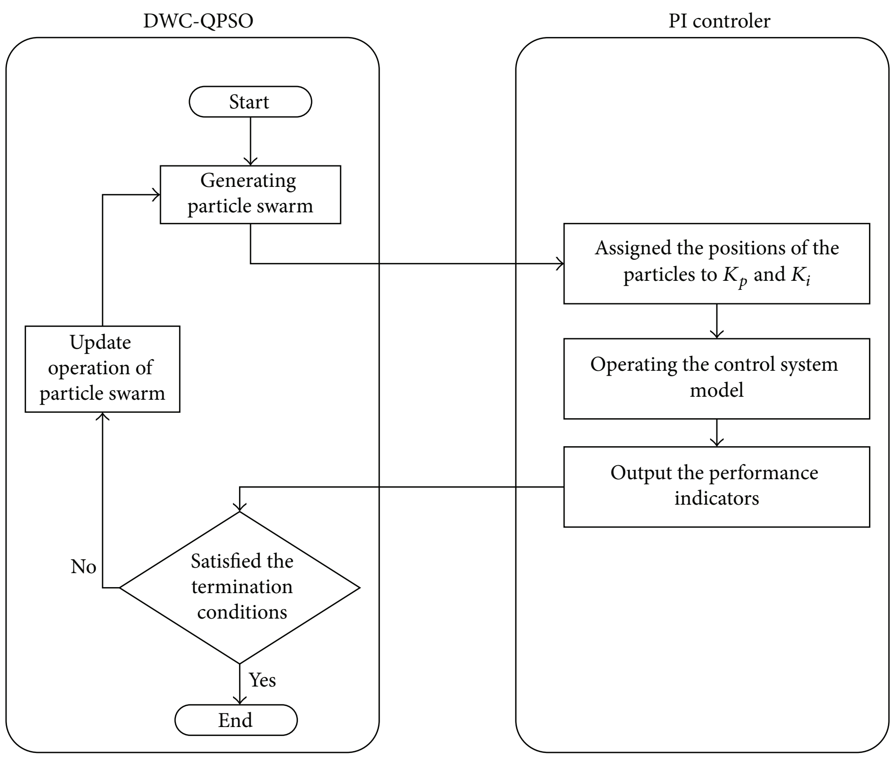

Figure 4: The process of optimizing on-ramp PI controller using DWC-QPSO algorithm.

Step 4. Calculate the mean best position $C(t)$ of the particle swarm based on (7), using it to evaluate $L_{i,j}(t)$, where i is the length of δ potential well.

Step 5. According to (9), update the location of particle i ($1 \le i \le N$).

Step 6. Recalculate the fitness value of the particle i's new position in the solution space according to the objective function.

Step 7. Update the locations of all particles' $P_i(t)$. The specific operation is comparing the fitness value of $X_i(t)$ and $P_i(t-1)$; if $f(X_i(t)) < f(P_i(t-1))$, do $P_i(t) = X_i(t)$; otherwise do $P_i(t) = P_i(t-1)$.

Step 8. Compare all particles' fitness value with $P_g(t-1)$'s fitness value. If $i \in S_1$ and $f(P_i(t)) < f(P_g(t-1))$, save i's position as the global best position of subgroup S_1, denoted by $P_{gS1}(t)$; otherwise do the operation of $P_g(t-1) \rightarrow P_{gS1}(t)$. If $i \in S_2$ and $f(P_i(t)) < f(P_g(t-1))$, save i's position as the global best position of subgroup S_2; otherwise do the operation of $P_g(t-1) \rightarrow P_{gS2}(t)$.

Step 9. Compare the fitness value of $P_{gS1}(t)$ with the fitness value of $P_{gS2}(t)$. If $f(P_{gS1}(t)) > f(P_{gS2}(t))$, $P_{gS2}(t)$ would be the global best position of the entire population $P_g(t)$; that is, $P_{gS2}(t) \rightarrow P_g(t)$; otherwise, do the operation of $P_{gS1}(t) \rightarrow P_g(t)$.

Step 10. Determine the termination conditions of the algorithm. If the end conditions are not reached (usually the end condition is the expected fitness value or the maximum number of iterations), do $t = t + 1$ and return to Step 3.

Step 11. Output the optimal solution.

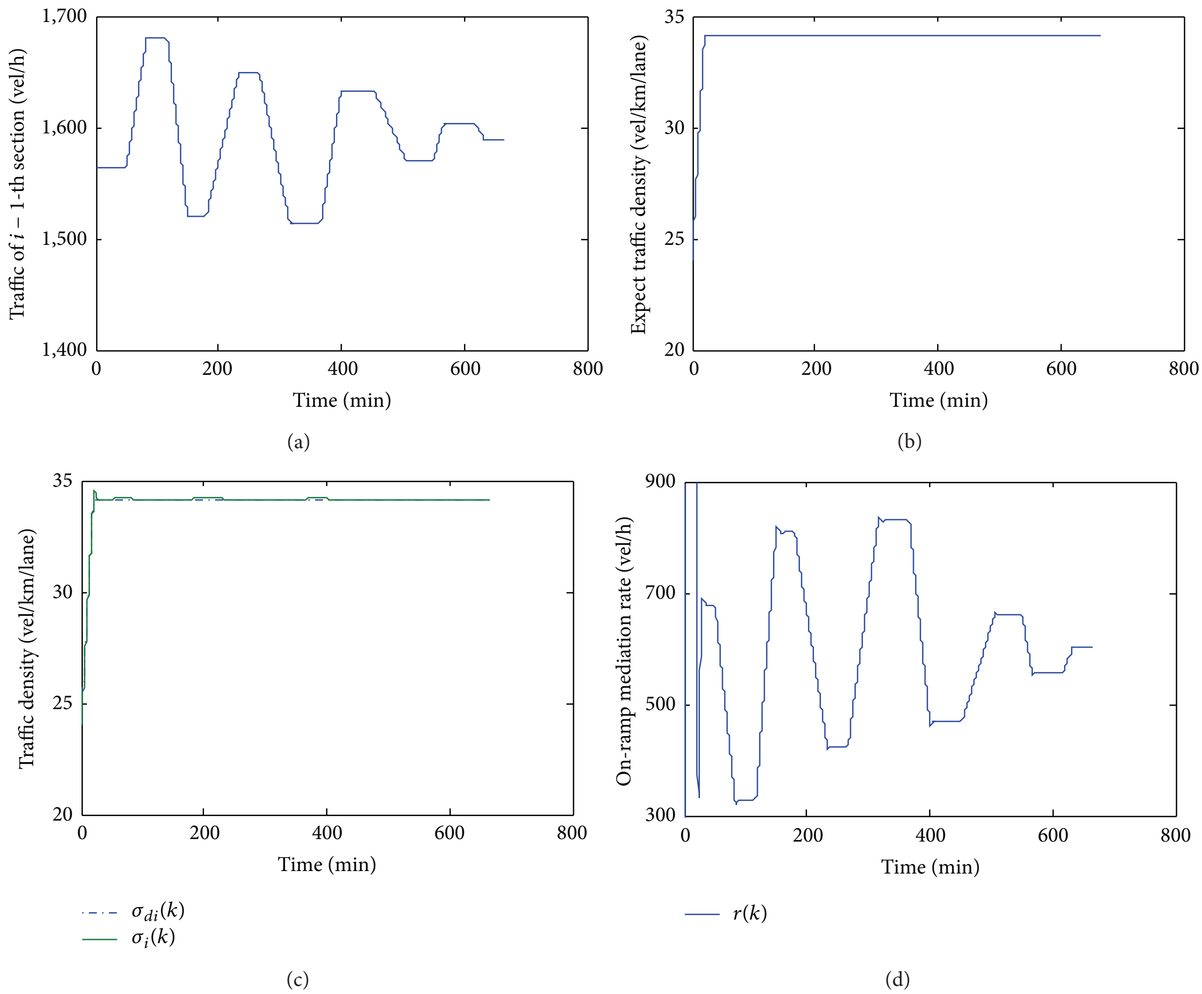

Figure 5: Comparison of actual and expected traffic density: (a) traffic flow of $i-1$ section; (b) expected traffic density; (c) traffic density comparison; (d) on-ramp mediation rate.

5. On-Ramp PI Controller Simulation and Results Analysis

In order to demonstrate the effectiveness of the new algorithm, an experiment for on-ramp PI controller based on DWC-QPSO algorithm is conducted. In the simulation, the acceleration coefficients of DWC-QPSO c_1 and c_2 are set to be 2, and the contraction-expansion factor α decreases linearly from 1 to 0.5. The maximum number of iteration steps *itemMax* is 150 and the entire population size N is 30. All particles are evenly distributed to two subgroups, which means that the population size of master subgroup N_{s1} is equal to the population size of secondary subgroup N_{s1}. The system prototype is implemented in MATLAB 7.9.0. Typical parameters of on-ramp traffic flow model are shown in Table 3.

Assume the curve of the traffic $q_{i-1}(k)$ is shown in Figure 5(a). And Figure 5(b) illustrates that the expected traffic density of ith section $\sigma_{di}(k)$ linearly increases from the initial 24.06 vel/km·lane^{-1} to 34.16 vel/km·lane^{-1}, and since then it has been maintained at 34.16 vel/km·lane^{-1}. The values of parameters of the on-ramp traffic PI controller optimized by DWC-PSO algorithm are $K_p = 186.6008$ and $K_i = 330.000$, which corresponds to the minimum fitness value $J = 2.054$. After using on-ramp traffic PI controller under the above parameter settings, the curves of actual traffic flow density of lane's ith section and the on-ramp mediation rate are shown in Figures 5(c) and 5(d), respectively.

Table 3: Parameters of on-ramp traffic flow model.

v_f	σ_{jam}	σ_c	λ	T
97.3 km/h	74 vel/km·lane^{-1}	37 vel/km·lane^{-1}	3	20 s

It is learned from Figure 5(c) that the error between $\sigma_i(k)$ and $\sigma_{di}(k)$ is very small, which means that the expected traffic density can be well tracked by the actual traffic density in the system using DWC-QPSO algorithm optimization. As can be seen from Figure 5(d), the on-ramp mediation rate increases with time at the beginning, and there is a surge during this period. Since the traffic of $(i-1)$th section of lane remains unchanged in the initial stage, while the expected traffic density is linearly growing from 24.06 vel/km·lane^{-1} to

Table 4: Parameters of PI controller and the objective function values.

Algorithms	K_P	K_i	J
DWC-QPSO	186.6008	330.000	0.2054
PSO [2]	184.4559	330.000	0.25083
ACO [1]	183.2129	329.0433	0.25198.

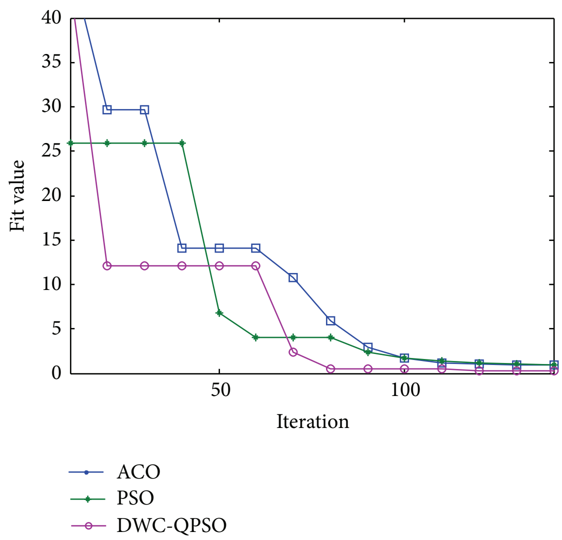

Figure 6: Convergence curve of algorithms.

34.16 vel/km·lane^{-1}, then traffic increment is mainly coming from the on-ramp. In the following stage, the expected traffic density remains at vel/km·lane^{-1}, while the on-ramp mediation rate changes with the traffic variation of $(i-1)$th section. The on-ramp mediation rate would be decreasing and the vehicles entering to the main lane from on-ramp would be reducing when the traffic of $i-1$ section is increasing; however, when the traffic of $i-1$ section is reducing, the on-ramp mediation rate would be increasing and the vehicles entering to the main lane from on-ramp would be increasing as well. The on-ramp control strategies above could avoid traffic congestion, keeping the traffic smooth and maximizing the utilization of the main lane.

In order to verify the effectiveness of the algorithm, the optimization performances of ant colony optimization (ACO) algorithm and particle swarm optimization (PSO) algorithm are compared with the performance of DWC-QPSO. The PI controller parameters and the optimal value of objective function corresponding to these three algorithms are shown in Table 4, and the curve of the fitness value is shown in Figure 6.

The simulation illustrates that the minimum fitness value of on-ramp PI controller system can be obtained by DWC-QPSO in these three algorithms. The results reveal that the actual traffic flow density $\sigma_i(k)$ is closest to the expected traffic density $\sigma_{di}(k)$ and the system has the highest stabilities under this condition. As a real-time traffic control system, if the reaction is too slow, the input information would be useless for the system and the outputs would lose effectiveness for inducing actual traffic. Therefore, the timeliness is an important evaluation standard for optimization algorithms in real-time applications. It can be learned that the DWC-QPSO algorithm has the fastest convergence speed comparing with the ant colony algorithm and the standard particle swarm optimization from Figure 6. Because the novel algorithm has high-speed response capabilities in avoiding cumbersome and complex human intervention, it is able to quickly find the optimal value of parameters K_p and K_i. In summary, because the system is capable of adjusting the on-ramp traffic dynamically in real-time according to the traffic condition of upper section, the DWC-QPSO algorithm is quite applicable in on-ramp traffic PI controller and outperforms other evolutionary algorithms with enhancement in both adaptability and stability.

6. Conclusion

On-ramp control is a significant way to control the traffic density of freeway. Furthermore, it is also an important foundation for the traffic guidance systems and other decision-making systems in ITS. In this paper, the on-ramp traffic PI controller based on the DWC-QPSO algorithm is proposed, which is designed on the assumption that all sections of lanes have the abilities of measuring the average vehicle speed and the traffic density. The DWC-QPSO algorithm is applied in optimizing the PI controller parameters K_p and K_i in order to achieve the traffic density control at the on-ramp. Experimental results show that the on-ramp traffic PI controller based on DWC-QPSO algorithm has better adaptability and stability compared to the other evolutionary algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Nature Science Foundation of China (no. 61104175) and Fundamental Research Funds for the Central Universities (no. 2014NZYQN24). Tao Wu, Yusong Yan, and Xi Chen would like to thank everyone who contributed towards the completion of this paper.

References

[1] Q. X. Lv and X. R. Liang, "Freeway on-ramp control based on ant colony algorithm," in *Measurement & Control Technology*, vol. 10, pp. 61–69, 2011.

[2] Y. K. Fan and X. R. Liang, "Particle swarm optimization based PI controller for freeway ramp metering," in *Proceedings of the 27th Chinese Control Conference*, pp. 503–506, Kunming, China, July 2008.

[3] J. X. Xu, S. H. Yu, and L. J. Xu, "Adaptive coordinated control based on FNN for freeway entering ramp," *Journal of Southeast University (Natural Science Edition)*, vol. S1, pp. 266–271, 2009.

[4] L. Lei, L.-Y. Dong, and H.-X. Ge, "Study of gear-alternating control regulation in the interfluent location of on-ramps with cellular automaton model," *Acta Physica Sinica*, vol. 56, no. 12, pp. 6874–6880, 2007.

[5] B. L. Li and S. R. Qu, "Adaptive control of on-ramp metering for highway based on genetic-fuzzy approach," *Measurement & Control Technology*, vol. 5, pp. 73–80, 2011.

[6] W.-S. Su, Y. Li, J.-G. Sun, and W. Zhou, "Optimization methodology of PI controller parameters for aeroengines based on genetic algorithm," *Journal of Aerospace Power*, vol. 20, no. 6, pp. 1078–1082, 2005.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

[8] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, IEEE Press, Nagoya, Japan, October 1995.

[9] Q.-J. Ni, Z.-Z. Zhang, Z.-Z. Wang, and H.-C. Xing, "Dynamic probabilistic particle swarm optimization based on varying multi-cluster structure," *Journal of Software*, vol. 20, no. 2, pp. 339–349, 2009.

[10] W. Jiao and G.-B. Liu, "Two subpopulation swarm PSO algorithm in dynamic environment," *Control and Decision*, vol. 24, no. 7, pp. 1083–1091, 2009.

[11] F. V. D. Bergh, *An analysis of particle swarm optimizers [Ph.D. thesis]*, University of Pretoria, 2002.

[12] J. Sun, *Particle Swarm Optimization with Particles Having Quantum Behavior*, Jiangnan University.

[13] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.