IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 5, 2020, accepted March 29, 2020, date of publication April 2, 2020, date of current version April 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985207

A Novel k-MPSO Clustering Algorithm for the Construction of Typical Driving Cycles

WEI YAN, MEI-JING LI, YONG-CHANG ZHONG, CHUN-YAN QU, AND GUO-XIANG LI
School of Energy and Power Engineering, Shandong University, Shandong 250061, China

Corresponding authors: Wei Yan (yanwei@sdu.edu.cn) and Guo-Xiang Li (liguox@sdu.edu.cn)

This work was supported in part by the National Key Research and Development Program, China, under Grant 2017YFB0103405, the Key Research and Development Program of Shandong Province, China, under Grant 2017CXGC0603, and in part by the Key Research and Development Program of Shandong Province, China, under Grant 2018GGX104020.

ABSTRACT The practical driving cycle is of great significance in studying the control strategy of vehicles, and effective clustering of micro-trips is the key to obtaining the typical driving cycle. A novel and efficient method for constructing typical driving cycles is presented in this paper. First, by combining the preying behavior and random behavior of the artificial fish swarm algorithm (AFSA) with particle swarm optimization (PSO), a modified particle swarm optimization (MPSO) is proposed. By comparing the means and standard deviations of the optima, MPSO is verified as much more accurate and stable than PSO, AFSA, select particle swarm optimization (SPSO) and cross particle swarm optimization (CPSO) in the optimization calculation of four typical multi-modal benchmark functions. Second, by applying MPSO to optimize the k-means algorithm, the k-MPSO clustering algorithm is obtained. In the case of clustering the Iris standard data set, the average error rates of the k-means algorithm and k-MPSO clustering algorithm are 11.6% and 7.8%, respectively, which means that the k-MPSO clustering algorithm has a stronger searching ability. Finally, with the ECAN Tools software, real-world driving data that include thousands of micro-trips in Jinan are collected, and 19 representative characteristic parameters are selected to fully describe the driving conditions. After principal component analysis (PCA), the k-MPSO clustering algorithm method is applied to cluster the micro-trips into three classes and construct the typical driving cycle in Jinan.

INDEX TERMS Artificial fish swarm algorithm, driving cycle, k-MPSO clustering algorithm, particle swarm optimization.

I. INTRODUCTION

To improve the fuel economy and emission characteristics of vehicles, the use of driving cycles in typical cities to establish appropriate vehicle control strategy has become a current research focus [1]–[3]. Certain standard driving cycles have been previously established, e.g., FTP72 of America and NEDC of Europe [4]. However, the driving conditions of different cities or vehicles are not consistent with the standard driving cycles. In recent years, many researchers [5]–[7] have made great efforts to study the driving cycles of typical cities or specific vehicles, e.g., a driving cycle for passenger cars and motorcycles in Chennai (India) [8], driving cycles for suburban road-work vehicles and airport vehicles [9], a dynamic driving cycle for public buses in Hamburg (Germany) [10], simulated driving cycles for light, medium, and heavy duty trucks in the Toronto Waterfront Area [11], and a driving cycle for a supercapacitor electric bus route in Hong Kong (China) [12].

A common approach to constructing driving cycles is the Micro-trips method [13]. However, the k-means clustering analysis used in the Micro-trips method is easily trapped in local optima when selecting the clustering centers. To obtain more accurate and effective clustering centers, it is necessary to improve the traditional k-means algorithm with metaheuristic search algorithms, e.g., artificial bee colony (ABC) [14], particle swarm optimization (PSO) [15] and firefly algorithm (FA) [16], which have validated the effectiveness of such hybrid clustering methods. Due to its simple updating formulas and excellent searching ability, PSO has been widely applied in clustering analysis [17] and other optimization fields, e.g., single- and multi-objective problems [18], artificial neural network improving problem [19], assignment problem [20], and optimum battery

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Munir.

energy storage problem [21]. However, the canonical PSO still must be improved with respect to parameter setting [22]–[24], updating strategy [25]–[28], and convergence theory [29]–[31], among others. Particularly, Wen *et al.* [32] introduced Gauss distribution function into PSO, and the obtained Gaussian-PSO had excellent optimization performance, which was also verified by Higashi and Iba [33] later.

In this paper, a modified particle swarm optimization (MPSO) is presented to improve the traditional k-means algorithm. Based on the idea of the artificial fish swarm algorithm (AFSA), the improvements to the PSO in this study focus on setting of the inertial weight and the updating strategy. In the optimization calculation of four typical multi-modal benchmark functions, the modified particle swarm optimization (MPSO) shows better performance in both local searching ability and global searching ability than PSO and AFSA. To further verify the effectiveness of MPSO, two other existing modified particle swarm optimizations, i.e., select particle swarm optimization (SPSO) and cross particle swarm optimization (CPSO) [34], are compared with MPSO in optimization calculation of the four test functions, proving that MPSO shows much better performance in both computational stability and accuracy. In clustering of the Iris standard data set, the average error rates of the k-means algorithm and k-MPSO clustering algorithm are 11.6% and 7.8%, respectively, which means that a stronger searching ability exists in the k-MPSO clustering algorithm. After collecting real-world driving data, selecting 19 representative characteristic parameters to fully describe the driving conditions, and reducing the dimensionality of the characteristic parameters matrix by principal component analysis (PCA), the k-MPSO clustering algorithm method is applied to cluster the thousands of micro-trips into three classes, from which the typical driving cycle in a typical city (Jinan, the capital of Shandong Province, China) is constructed.

The remainder of this paper is organized as follows. In Section II, the k-MPSO clustering algorithm method is presented and analyzed in detail. The detailed process of constructing the driving cycle in Jinan is described in Section III. Finally, in Section IV, the conclusions from this study with respect to the k-MPSO clustering algorithm method are presented.

II. PROPOSED k-MPSO CLUSTERING ALGORITHM

A. DESCRIPTION OF THE PSO PRINCIPLE

Proposed by Kennedy and Eberhart in 1995 [35], the canonical PSO assumes the existence of a bird flock with a limited number of birds that search for food in a given solution space. The exact locations of the food are unknown to all birds, but the distances between the birds and the food are known. The most efficient way to find the food is to search near the bird that is closest to the food.

Assuming that a particle swarm exists with m number of particles, the position and velocity of particle i are represented as $x_i = (x_{i1}, x_{i2}, \ldots, x_{is})$ and $v_i = (v_{i1}, v_{i2}, \ldots, v_{is})$, respectively, where $i = 1, 2, \ldots, m$, and s is the dimensionality of particle i. The personal best solution found so far by itself is represented as $p_i = (p_{i1}, p_{i2}, \ldots, p_{is})$, and the global best solution found by the entire particle swarm is represented as $p_g = (p_{g1}, p_{g2}, \ldots, p_{gs})$. In the iteration process, each particle updates its position and velocity as follows:

$$v_i^{r+1} = w \cdot v_i^r + c_1 \cdot rand \cdot (p_i^r - x_i^r) + c_2 \cdot rand \cdot (p_g^r - x_i^r) \tag{1}$$

$$x_i^{r+1} = x_i^r + v_i^{r+1} \tag{2}$$

where w is the inertial weight, r is the *rth* iteration, c_1 and c_2 are positive acceleration constants, normally $c_1 = c_2 = 1.4995$, and *rand* is a random sequence in the range of [0,1]. To prevent the particle from searching outside the solution space, the position and velocity are limited within the ranges of $[x_{min}, x_{max}]$ and $[v_{min}, v_{max}]$, respectively.

B. MODIFIED PSO

In the canonical PSO, the particle searches to approach the personal and global best solution, which makes it converge quickly but easily fall into local optima, thus displaying the premature phenomenon. Thus, an improved method of inertia weight is proposed, and the idea of AFSA is introduced into the particle search strategy.

1) IMPROVEMENT OF THE INERTIAL WEIGHT

The inertial weight w represents the ability of particles to maintain self-motion inertia. Earlier researchers set this weight to a fixed value of 1.0 [35], which makes the speed of convergence too rapid to obtain sufficient local searching performance in the late iteration stages.

A better global searching ability and a worse local searching ability exist with a larger inertia weight. Thus, an improved method for inertia weight is proposed. The inertia weight varies slowly at the early iteration stage and decreases linearly at the later stage, thus producing a strong global searching ability as well as a strong local searching ability in the entire iteration stage. The variation of inertia weight is shown in Fig. 1 and the following equation.

$$w = \begin{cases} w_{max} \\ -(w_{max} - w_{min}) \cdot (\dfrac{r}{R})^2, & 0 < r \le \dfrac{1}{2}R \\ \dfrac{3}{2}w_{max} - \dfrac{1}{2}w_{min} \\ -\dfrac{3}{2}(w_{max} - w_{min}) \cdot \dfrac{r}{R}, & \dfrac{1}{2}R < r \le R \end{cases} \tag{3}$$

where w_{max} and w_{min} are the maximum and minimum of w, which are set to 0.9 and 0.4, respectively, and R is the total number of the iterations.

2) IMPROVEMENT OF THE PARTICLE UPDATE STRATEGY

AFSA is also a swarm intelligence algorithm based on animal behaviors in which swarming and following behavior are performed to search for the global optimal value, and preying and random behavior are performed to search for the local optimal value. To strengthen the local searching ability in

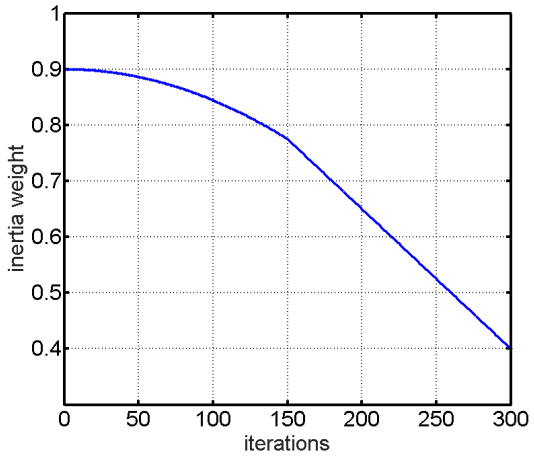


FIGURE 1. The variation of inertia weight.

the early stage, the preying behavior and random behavior of AFSA are introduced into the PSO.

Preying behavior: Each particle of the PSO is treated as an artificial fish, and its velocity is treated as the visual of the artificial fish. The particle x_i attempts to find the location x_{i0} with higher food concentration in its visual range as follows:

$$x_{i0} = x_i + rand \cdot v_i \tag{4}$$

If x_{i0} exists, the velocity and position of x_i are updated respectively as follows:

$$v_i^{r+1} = w \cdot v_i^r + c_1 \cdot rand \cdot (x_{i0} - x_i^r) + c_2 \cdot rand \cdot (p_g^r - x_i^r) \tag{5}$$

$$x_i^{r+1} = x_i^r + v_i^{r+1} + step \cdot rand \cdot \frac{x_{i0} - x_i^r}{||x_{i0} - x_i^r||} \tag{6}$$

where v_i is the velocity of the particle, which also the visual of the artificial fish, and *step* is the moving step of the particle.

If x_{i0} cannot be found after attempting the set times (*trynumber*), the random behavior is performed.

Random behavior: This behavior is different from that of traditional AFSA. All particles with fitness value greater than the particle x_i are found. One of the particles, x_j, is randomly selected as the direction to update the velocity and position of particle x_i, which can be expressed as follows:

$$v_i^{r+1} = w \cdot v_i^r + c_1 \cdot rand \cdot (x_j - x_i^r) + c_2 \cdot rand \cdot (p_g^r - x_i^r) \tag{7}$$

$$x_i^{r+1} = x_i^r + v_i^{r+1} + step \cdot rand \cdot \frac{x_j - x_i^r}{||x_j - x_i^r||} \tag{8}$$

At the early iteration stage, the particle swarm often misses the global optimum and falls into a local optimum due to the large inertia weight. Thus, it is necessary to strengthen the local searching ability in the early stage. The particle swarm performs preying behavior first. It can be observed that the historical optimal position of each particle is replaced by the better position in its visual range, which strengthens the local searching ability of the particle. At the later iteration stage, as the inertia weight decreases, the local searching ability strengthens, and the global searching ability weakens. Therefore, the velocity and position of each particle are updated according to the canonical PSO, which improves the global searching ability. MPSO guarantees the global searching ability and strengthens its local searching ability at both the early and later stages. The process of MPSO is given as shown in Fig. 2.

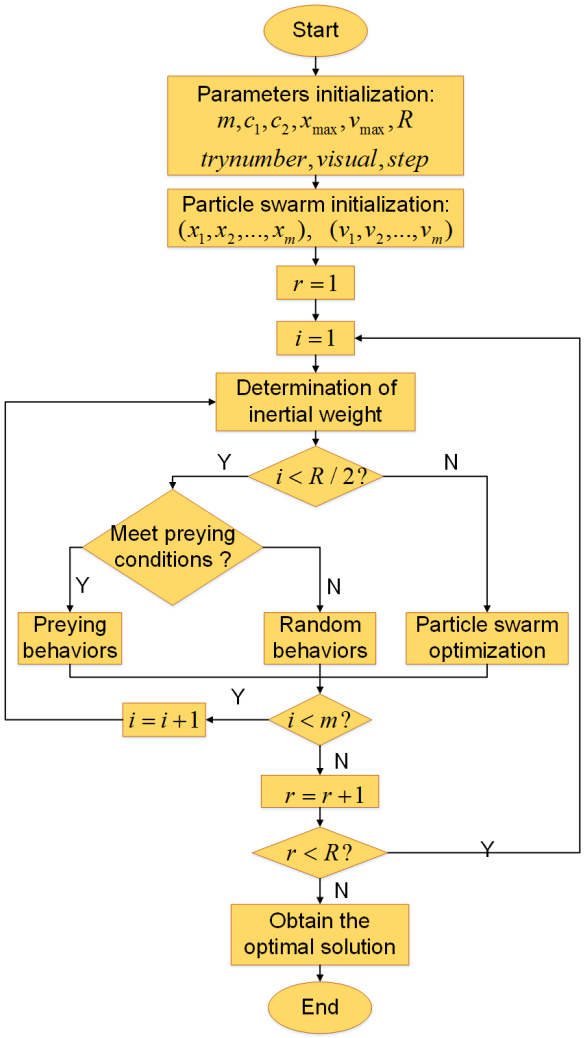


FIGURE 2. The flow diagram of MPSO.

3) VERIFICATION OF THE EFFECTIVENESS OF MPSO

To verify the effectiveness of the proposed MPSO, 4 typical multi-modal benchmark functions are selected to test the performance. The four functions are shown in (9) through (12) and Fig. 3, and there is only one global optimum but many local extrema for each function.

$$F_1(x, y) = -20 \cdot e^{-0.2 \cdot \sqrt{\frac{x^2+y^2}{2}}} - e^{\frac{cos2\pi x + cos2\pi y}{2}} + 20 + e \tag{9}$$

TABLE 1. The comparison of experimental results on 4 test functions.

Function	Opt	Algorithm	Best	Worst	Mean	Std
F1	0	PSO	3.778E-4	4.173E-3	1.378E-3	9.984E-4
		AFSA	1.139E-4	4.690E-4	2.662E-4	1.017E-4
		SPSO	2.625E-6	4.716E-4	1.316E-4	1.227E-4
		CPSO	6.128E-14	1.956E-8	1.876E-9	4.848E-9
		MPSO	8.882E-16	1.510E-14	3.553E-15	3.235E-15
F2	1.0054	PSO	1.0054	0.9243	1.0013	1.812E-2
		AFSA	1.0054	1.0046	1.0052	2.161E-4
		SPSO	1.0054	0.9244	1.0001	1.869E-2
		CPSO	1.0054	0.9243	0.9973	2.439E-2
		MPSO	1.0054	1.0054	1.0054	4.556E-16
F3	0	PSO	5.869E-5	1.913E-2	3.327E-3	4.376E-3
		AFSA	1.340E-5	6.345E-3	1.710E-3	1.867E-3
		SPSO	7.216E-16	1.913E-2	3.827E-3	7.852E-3
		CPSO	7.396E-11	1.913E-2	8.674E-3	9.708E-3
		MPSO	5.551E-17	9.992E-16	5.773E-16	2.967E-16
F4	0	PSO	3.244E-7	9.867E-3	3.473E-3	3.941E-3
		AFSA	3.180E-7	9.871E-3	6.781E-3	3.276E-3
		SPSO	5.551E-16	7.396E-3	2.842E-3	3.606E-3
		CPSO	1.366E-14	9.865E-3	2.651E-3	3.631E-3
		MPSO	4.441E-16	9.659E-15	5.235E-15	2.821E-15

Note: "Opt" is the known optimal solution of each test function, and "Std" is the standard deviation of the obtained optimal values for the 20 independent runs.

where x and y are limited within the range of [-5,5].

$$F_2(x,y) = \frac{sin\sqrt{\frac{x^2+y^2}{2}}}{\sqrt{\frac{x^2+y^2}{2}}} + e^{\frac{cos2\pi x+cos2\pi y}{2}} - e \tag{10}$$

where x and y are limited within the range of [-2,2].

$$F_3(x,y) = \frac{sin^2\sqrt{\frac{x^2+y^2}{2}} - 0.5}{[1+0.001(x^2+y^2)]^2} + 0.5 \tag{11}$$

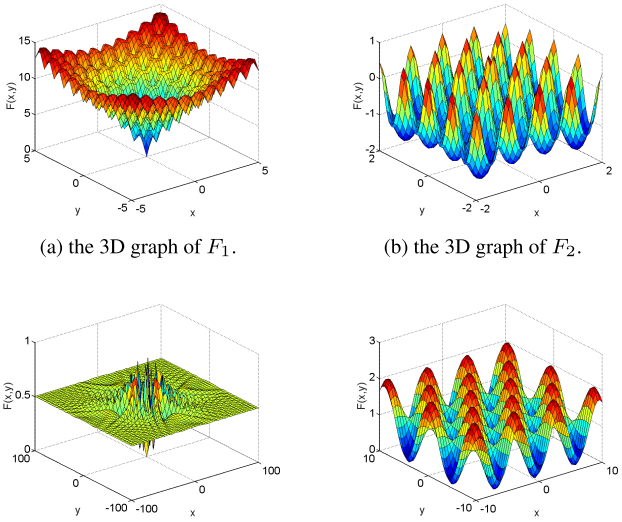
where x and y are limited within the range of [-100,100].

$$F_4(x,y) = \frac{x^2+y^2}{4000} - cosx \cdot cos\frac{y}{\sqrt{2}} + 1 \tag{12}$$

where x and y are limited within the range of [-10,10].

All algorithms are run on a PC with a 3.20 GHz quad-core processor and 16.0 GB RAM. With the same MATLAB coding environment, each function is calculated 20 times with PSO, AFSA, SPSO, CPSO and MPSO. The experiments attempt to ensure that the parameters of each algorithm are consistent: the total number of iterations is 300, and the number of particles or fish is 40. The parameters of AFSA are set as follows: *visual* is 0.5 times the independent variable upper limit for each test function, and $step = 0.4visual$. The parameters of MPSO are set as follows: $c_1 = c_2 = 1.4995$, v_{max} is 0.2 times the independent variable upper limit for each test function, $visual = v$, $step = 0.5visual$, and $trynumber = 100$. The inertia weights of SPSO and CPSO are set as 0.9, and the other parameters are the same as those of MPSO. The experimental results of each algorithm are compared as shown in Table 1.

It can be observed that MPSO is much better than the other algorithms in both computational stability and accuracy under the same test environment and parameter settings, which shows that the improved method is effective in searching for optimal values.

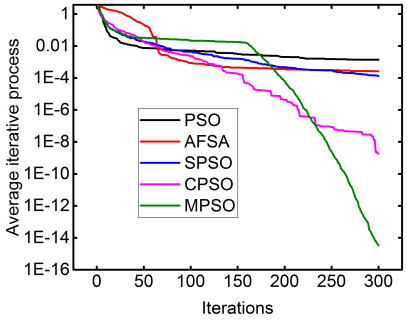


(a) the 3D graph of F_1. (b) the 3D graph of F_2.

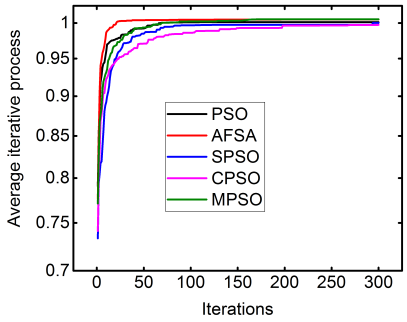
(c) the 3D graph of F_3. (d) the 3D graph of F_4.

FIGURE 3. The three-dimensional graphs of the four test functions.

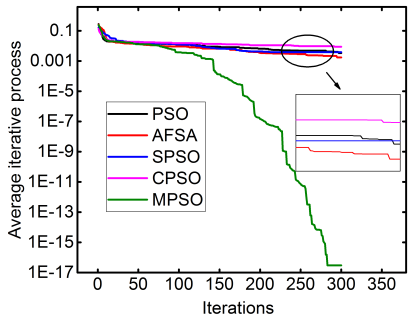
To further compare the performances, Fig. 4 shows the average iteration process of 20 runs for each algorithm. MPSO has a higher convergence accuracy than the other algorithms. In optimization of each function, the convergence speed of the other algorithms might be faster in the early stage, but the optimal values are no longer updated while the iteration continues. Due to the introduction of preying and random behaviors, MPSO tends towards convergence quickly while ensuring exploration in the full search space in the early stage. With the decrease in the inertia weight, the local searching ability is enhanced and converges, further



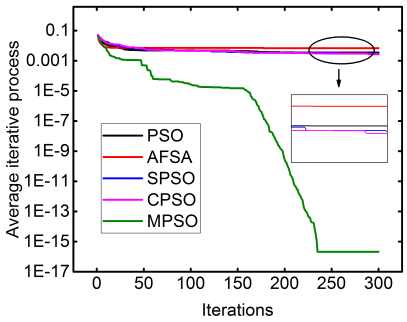
(a) The average iteration processes of F_1.



(b) The average iteration processes of F_2.



(c) The average iteration processes of F_3.



(d) The average iteration processes of F_4.

FIGURE 4. Comparison of the average iteration processes for each algorithm.

TABLE 2. Clustering results comparison of Iris standard data set.

	k-means	k-MPSO
Numbers of falling into local optimum	2	0
Maximum number of errors	18	16
Minimum number of errors	15	11
Average number of errors	17.4	11.8
Average error rate	11.6%	7.81%

improving the accuracy over that of PSO, AFSA, SPSO and CPSO.

C. PROPOSED k-MPSO CLUSTERING ALGORITHM

The k-MPSO clustering algorithm is a hybrid clustering algorithm based on MPSO and the k-means clustering algorithm, for which the position of each particle represents the position of k clustering centers. The mathematical process is shown as follows:

Suppose there is a data set Y with a number of h groups of data, which can be written as $Y = (y_{l1}, y_{l2}, \ldots, y_{ln})$, where $l = 1, 2, \ldots, h$.

First, the position x_i and velocity v_i of the particle i are initialized randomly as follows: $x_i = (c_1, c_2, \ldots, c_k)$ and $v_i = (q_1, q_2, \ldots, q_k)$, where $c_j = (c_{j1}, c_{j2}, \ldots, c_{jn})$, $q_j = (q_{j1}, q_{j2}, \ldots, q_{jn})$, $i = 1, 2, \ldots, m$ and m particles cluster the entire data set in m different ways, $j = 1, 2, \ldots, k$, and c_j and q_j are the jth clustering center and its velocity, respectively.

The data set Y are clustered into k classes according to the Euclidean distance from each data y_l to each cluster center c_j of the particle i. The sum of distances is calculated to evaluate the fitness value of the particle i, which is shown in (13).

$$f(x_i) = \sum_{j=1}^{k} \sum_{l=1}^{h} \sqrt{\sum_{o=1}^{n} (y_{lo} - c_{jo})^2} \tag{13}$$

where c_j is the closest clustering center to y_l.

When the individual and global optimal values of the initial generation are determined, the positions and velocities of the m particles are updated to obtain the minimum fitness value $f_{min}(x_a)$ based on the proposed MPSO. After obtaining the optimal clustering centers according to the particle a, the data set Y is eventually clustered into k classes.

The Iris standard data set is selected to compare the performance of k-MPSO with that of the k-means clustering algorithm. There are four attributes and a total of 150 groups of data to be divided into three classes in the data set. All data are standardized by the min-max function before clustering. Each algorithm is run 20 times to obtain the clustering centers, and the results are shown in Table 2. It can be observed that the k-MPSO clustering algorithm has a lower average error rate than the k-means clustering algorithm. Furthermore, k-means analysis also easily falls into local optima, and the efficiency of the k-MPSO clustering algorithm is strengthened, which means that it can be used in application and research of practical engineering problems.

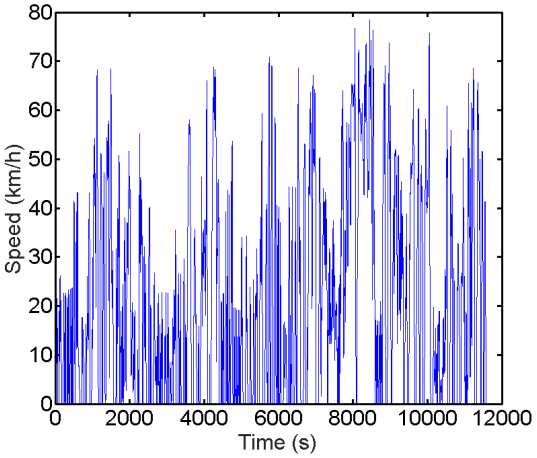


FIGURE 5. The speed-time profile of a trip.

III. CONSTRUCTION OF A TYPICAL DRIVING CYCLE

A. DATA ACQUISITION AND PROCESSING

Within the main urban areas in Jinan, the testers drive vehicles according to their purpose, without a given route or date, to obtain driving data for the period 7:00-9:00. The data acquisition device is the USBCAN-OBD, which collects data through the controller area network (CAN) on vehicles connected with the OBD interface. Using the ECAN Tools software, the data frames obtained through CAN are identified by the software OBD interface to obtain the parameters, e.g., the engine speed and vehicle speed. To obtain sufficient data to construct a typical driving cycle, the experimental vehicles drive to different destinations as many times as possible. Fig. 5 shows the speed-time profile of one trip.

In the testing process, abnormal points occasionally appear in the collected data, which can affect the entire speed-time profile. Therefore, first, the abnormal points are removed by a filtering method as follows:

$$v_t = \begin{cases} 0, & v_{t-1} + v_{t+1} = 0 \\ 0, & v_{t-2} + v_{t+2} = 0 \\ v_t, & else \end{cases} \tag{14}$$

Second, the speed curve is corrected by a high-frequency interference filter as follows [36]:

$$v_t = \frac{1}{T} \cdot \sum_{x=-T}^{T} K(\frac{x}{T}) \cdot v_{t+x} \tag{15}$$

where v_t is the vehicle speed at the moment t, $T = 4s$, and $K(\frac{x}{T})$ represents the weightiness of the vehicle speed at the moment $t + x$, which can be obtained via the following equation[37].

$$K(\frac{x}{T}) = \begin{cases} \frac{T^2 - 1}{T^2} \cdot [1 - (\frac{x}{T})^2]^2, & \frac{x}{T} < 1 \\ 0, & else \end{cases} \tag{16}$$

Affected by many factors, e.g., traffic stream, pedestrians and traffic lights, driving vehicles show different kinematic modes. Thus, 4 kinematic modes are defined as follows:

TABLE 3. The 19 representative characteristic parameters.

Order	Parameters
1	Average driving speed(km/h)
2	Average speed(km/h)
3	Standard deviation of speed(km/h)
4	Maximum speed(km/h)
5	Average acceleration(m/s^2)
6	Maximum acceleration(m/s^2)
7	Standard deviation of acceleration(m/s^2)
8	Average deceleration(m/s^2)
9	Maximum deceleration(m/s^2)
10	Standard deviation of deceleration(m/s^2)
11	Time of the micro-trip(s)
12	Time of idling mode(s)
13	Time of acceleration mode(s)
14	Time of cruising mode(s)
15	Time of deceleration mode(s)
16	Time percentage of idling mode(-)
17	Time percentage of acceleration mode(-)
18	Time percentage of cruising mode(-)
19	Time percentage of deceleration mode(-)

1. Idling mode: $v = 0$ and $a = 0$,
2. Acceleration mode: $v > 0$ and $a \geq 0.15$,
3. Cruising mode: $v > 0$ and $|a| < 0.15$,
4. Deceleration mode: $v > 0$ and $a \leq -0.15$,

where v is the vehicle speed (m/s), and a is the acceleration (m/s^2).

The speed-time data are divided into diverse kinematic micro-trips from the last idling mode to the current idling mode. To describe the kinematic micro-trips accurately and comprehensively, 19 representative characteristic parameters are selected and are shown in Table 3.

By inputting the collected speed-time data, characteristic parameters of 108 kinematic micro-trips are extracted from thousands of micro-trips, which can be shown as Table 4.

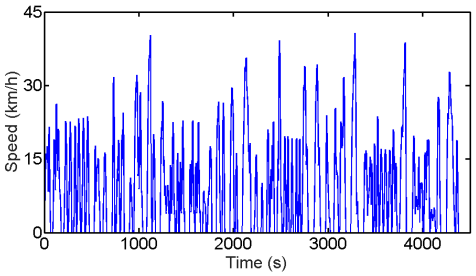
In the process of analyzing the driving data, the information on the kinematic micro-trips might be similar because too many characteristic parameters are selected. Thus, the principal component analysis (PCA) method [38], [39] is adopted to reduce the dimensionality of the characteristic parameter matrix. First, the obtained characteristic matrix is standardized to eliminate the influence of the magnitude order and unit. Via mathematical transformation, several principle components with a lack of linear correlation are extracted from the characteristic parameters matrix to reflect the original characteristic parameters more comprehensively. By calculation, the eigenvalues of the first three principle components are 8.618, 4.213 and 2.458, respectively, of which the accumulative contribution rate is 80.466%, i.e., they are sufficient to describe the 108 kinematic micro-trips.

B. CLUSTERING ANALYSIS BASED ON k-MPSO

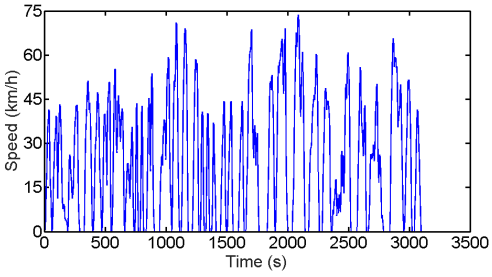
By reducing the dimensionality of the original characteristic matrix to obtain a three-dimensional matrix, the computational complexity is greatly decreased, and the k-MPSO clustering algorithm is applied to cluster the 108 kinematic micro-trips.

TABLE 4. The characteristic parameters of 108 kinematic micro-trips.

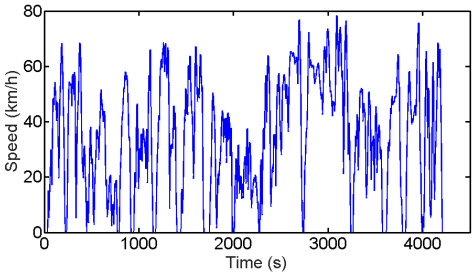
Order	Average driving speed(km/h)	Average speed(km/h)	...	Time percentage of cruising mode(-)	Time percentage of deceleration mode(-)
1	12.18	11.25	...	27%	32%
2	13.69	10.15	...	16%	34%
3	11.96	6.16	...	3%	22%
...	...	...	...	...	...
106	41.84	27.81	...	18%	27%
107	29.52	26.10	...	12%	34%
108	22.84	15.74	...	1%	38%



(a) The first class.



(b) The second class.



(c) The third class.

FIGURE 6. The clustering result of 108 kinematic micro-trips.

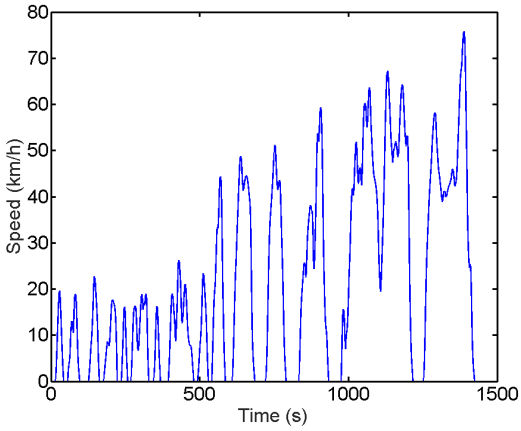
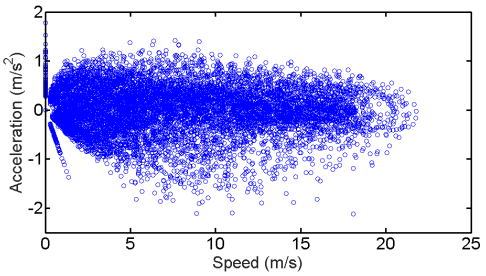
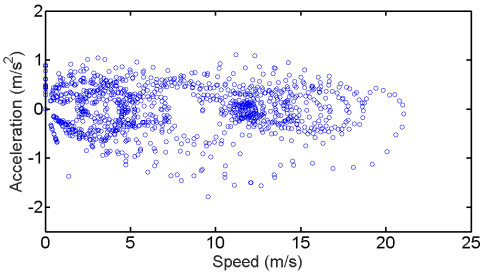


FIGURE 7. The typical driving cycle in Jinan.



(a) The distribution of the 108 kinematic micro-trips.



(b) The distribution of the typical driving cycle.

FIGURE 8. The distribution of acceleration-speed scatters.

As shown in Fig. 6, the samples are divided into three classes to represent the low-speed, medium-speed and high- speed conditions of Jinan. The first class consists of low-speed conditions, representing urban crowded traffic. Frequent modes of rapid acceleration and deceleration occur, with a short cruising time of the micro-trips and a high percentage of idling time. The second class consists of medium-speed conditions, for which the maximum speeds fall mostly in the range of 30-60 km/h, the idling modes are relatively small, traffic is not heavy, and acceleration and deceleration modes often occur. The third class consists of high-speed conditions, for which the maximum speeds exceed 60 km/h, the average speed is high, and the driving time is long with a high percentage of cruising time, i.e., good traffic conditions. It can be inferred that the first class represents urban driving conditions on residential roads, the second class represents minor arterial roads, and the third class represents arterial roads.

C. CONSTRUCTION OF THE DRIVING CYCLE

Referring to the research in [40], the total time of the constructed driving cycle is set to approximately 1500 s. After calculating the percentage of the time length of the above three classes, the time allocation of each class in the constructed driving cycle can be determined. The kinematic micro-trips with characteristic parameters closest to the clustering centers are selected to represent the typical driving conditions of this class. The typical driving cycle in Jinan is shown in Fig. 7.

To verify the effectiveness of the typical driving cycle, acceleration-speed scatter diagrams are constructed to describe the distribution before and after clustering and are shown in Fig. 8.

As shown in Fig. 8, the distribution of the typical driving cycle is similar to that of the original data, i.e., it can represent the typical driving conditions in Jinan and can be applied to establish an appropriate vehicle control strategy.

IV. CONCLUSION

To improve the fuel economy and emission characteristics of vehicles, it is of great importance to construct a practical driving cycle in a specific city for study of vehicle control strategies. A novel and efficient method for constructing typical driving cycles is presented in this paper.

Effective clustering of micro-trips is the key to obtaining the typical driving cycle, and thus, the k-MPSO clustering algorithm is presented. First, by combining the preying behavior and random behavior of AFSA with the PSO method, a modified particle swarm optimization (MPSO) approach is proposed. MPSO is demonstrated to produce much better accuracy and stability than PSO, AFSA, SPSO and CPSO in comparison of the optima means and standard deviations for optimization calculation of four typical multi-modal benchmark functions. Second, by applying MPSO to optimize the k-means algorithm, the k-MPSO clustering algorithm is obtained. Compared with the k-means clustering algorithm, the k-MPSO clustering algorithm shows better performance with a lower average error rate of 7.81% compared with the average error rate of 11.6% in the case of clustering the Iris standard data set.

The proposed k-MPSO clustering algorithm is applied to construction of the typical driving cycle in Jinan. Using PCA and the k-MPSO clustering algorithm method, the collected data are clustered into three classes that reflect with the real-world driving conditions of vehicles in Jinan. The distribution of the acceleration-speed scatters also proves that the typical driving cycle can reflect the traffic characteristics of Jinan to a certain extent. In conclusion, the proposed method is a feasible and efficient way to apply the proposed k-MPSO clustering algorithm to construction of the typical driving cycle.

REFERENCES

[1] S. Stockar, V. Marano, M. Canova, G. Rizzoni, and L. Guzzella, "Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 2949–2962, Sep. 2011.

[2] S.-H. Ho, Y.-D. Wong, and V. W.-C. Chang, "Developing Singapore driving cycle for passenger cars to estimate fuel consumption and vehicular emissions," *Atmos. Environ.*, vol. 97, pp. 353–362, Nov. 2014.

[3] K. Hereijgers, E. Silvas, T. Hofman, and M. Steinbuch, "Effects of using synthesized driving cycles on vehicle fuel consumption," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7505–7510, Jul. 2017.

[4] J. Lin, "A Markov process approach to driving cycle development," Ph.D. dissertation, Dept. Civil Environ. Eng., Univ. California, Los Angeles, CA, USA, 2002.

[5] P. Nyberg, E. Frisk, and L. Nielsen, "Using real-world driving databases to generate driving cycles with equivalence properties," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4095–4105, Jun. 2016.

[6] F. Millo, L. Rolando, R. Fuso, and J. Zhao, "Development of a new hybrid bus for urban public transportation," *Appl. Energy*, vol. 157, pp. 583–594, Nov. 2015.

[7] J. Brady and M. O'Mahony, "Development of a driving cycle to evaluate the energy economy of electric vehicles in urban areas," *Appl. Energy*, vol. 177, pp. 165–178, Sep. 2016.

[8] N. H. Arun, S. Mahesh, G. Ramadurai, and S. M. S. Nagendra, "Development of driving cycles for passenger cars and motorcycles in Chennai, India," *Sustain. Cities Soc.*, vol. 32, pp. 508–512, Jul. 2017.

[9] P. Seers, G. Nachin, and M. Glaus, "Development of two driving cycles for utility vehicles," *Transp. Res. D, Transp. Environ.*, vol. 41, pp. 377–385, Dec. 2015.

[10] R. Günther, T. Wenzel, M. Wegner, and R. Rettig, "Big data driven dynamic driving cycle development for busses in urban public transportation," *Transp. Res. D, Transp. Environ.*, vol. 51, pp. 276–289, Mar. 2017.

[11] G. Amirjamshidi and M. J. Roorda, "Development of simulated driving cycles for light, medium, and heavy duty trucks: Case of the Toronto waterfront area," *Transp. Res. D, Transp. Environ.*, vol. 34, pp. 255–266, Jan. 2015.

[12] H. Y. Tong, "Development of a driving cycle for a supercapacitor electric bus route in Hong Kong," *Sustain. Cities Soc.*, vol. 48, Jul. 2019, Art. no. 101588.

[13] J. Zhang, Z. Wang, P. Liu, Z. Zhang, X. Li, and C. Qu, "Driving cycles construction for electric vehicles considering road environment: A case study in Beijing," *Appl. Energy*, vol. 253, Nov. 2019, Art. no. 113514.

[14] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial bee colony (ABC) algorithm," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 652–657, Jan. 2011.

[15] A. Bouyer and A. Hatamlou, "An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms," *Appl. Soft Comput.*, vol. 67, pp. 172–182, Jun. 2018.

[16] H. Xie, L. Zhang, C. P. Lim, Y. Yu, C. Liu, H. Liu, and J. Walters, "Improving K-means clustering with enhanced firefly algorithms," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105763.

[17] V. Mangat, "Survey on particle swarm optimization based clustering analysis," in *Swarm and Evolutionary Computation*, L. Rutkowski, Ed. Berlin, German: Springer-Verlag, 2012, p. 301–309.

[18] M. Das and S. Dehuri, "Some studies on particle swarm optimization for single and multi-objective problems," in *Integration of Swarm Intelligence and Artificial Neural Network*, vol. 78, Singapore: World Scientific, 2011, pp. 239–304.

[19] Y. Wang, H. Liu, Z. Yu, and L. Tu, "An improved artificial neural network based on human-behaviour particle swarm optimization and cellular automata," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112862, doi: 10.1016/j.eswa.2019.112862.

[20] R. F. Abdel-Kader, "An improved PSO algorithm with genetic and neighborhood-based diversity operators for the job shop scheduling problem," *Appl. Artif. Intell.*, vol. 32, no. 5, pp. 433–462, May 2018.

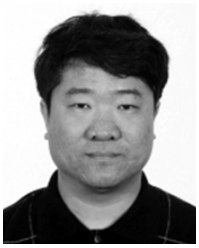
[21] T. Kerdphol, Y. Qudaih, and Y. Mitani, "Optimum battery energy storage system using PSO considering dynamic demand response for microgrids," *Int. J. Elec. Power*, vol. 83, pp. 58–66, Dec. 2016.

[22] K. M. Ang, W. H. Lim, N. A. M. Isa, S. S. Tiang, and C. H. Wong, "A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112882, doi: 10.1016/j.eswa.2019.112882.

[23] E. T. Oldewage, A. P. Engelbrecht, and C. W. Cleghorn, "Movement patterns of a particle swarm in high dimensional spaces," *Inf. Sci.*, vol. 512, pp. 1043–1062, Feb. 2020, doi: 10.1016/j.ins.2019.09.057.

[24] M. Isiet and M. Gadala, "Self-adapting control parameters in particle swarm optimization," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105653.

[25] S. M. Elsayed, R. A. Sarker, and E. Mezura-Montes, "Self-adaptive mix of particle swarm methodologies for constrained optimization," *Inf. Sci.*, vol. 277, pp. 216–233, Sep. 2014.
[26] E. Naderi, M. Pourakbari-Kasmaei, and H. Abdi, "An efficient particle swarm optimization algorithm to solve optimal power flow problem integrated with FACTS devices," *Appl. Soft Comput.*, vol. 80, pp. 243–262, Jul. 2019.
[27] T.-H.-S. Li, C.-Y. Liu, P.-H. Kuo, N.-C. Fang, C.-H. Li, C.-W. Cheng, C.-Y. Hsieh, L.-F. Wu, J.-J. Liang, and C.-Y. Chen, "A three-dimensional adaptive PSO-based packing algorithm for an IoT-based automated e-Fulfillment packaging system," *IEEE Access*, vol. 5, pp. 9188–9205, 2017.
[28] T. Blackwell and J. Kennedy, "Impact of communication topology in particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 689–702, Aug. 2019.
[29] S. A. Hosseini, A. Hajipour, and H. Tavakoli, "Design and optimization of a CMOS power amplifier using innovative fractional-order particle swarm optimization," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105831, doi: 10.1016/j.asoc.2019.105831.
[30] M. Kohler, M. M. B. R. Vellasco, and R. Tanscheit, "PSO+: A new particle swarm optimization algorithm for constrained problems," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105865, doi: 10.1016/j.asoc.2019.105865.
[31] M. Clerc and J. Kennedy, "The particle swarm–explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
[32] Z. Wen, *Proficient in MATLAB Intelligent Algorithm*. Beijing, China: Tsinghua Univ., 2015, pp. 130–135.
[33] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation," in *Proc. IEEE Swarm Intell. Symp.*, Indianapolis, IN, USA, Apr. 2003, pp. 72–79.
[34] N. Sinha and B. Purkayastha, "PSO embedded evolutionary programming technique for nonconvex economic load dispatch," in *Proc. IEEE PES Power Syst. Conf. Expo.*, New York, NY, USA, Oct. 2004, pp. 66–71.
[35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE 1st Int. Conf. Neural Netw.*, Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.
[36] Y.-J. Zhang and Y.-B. Da, "The decomposition of energy-related carbon emission and its decoupling with economic growth in China," *Renew. Sustain. Energy Rev.*, vol. 41, pp. 1255–1266, Jan. 2015.
[37] Q. Zhu and T. Wei, "Household energy use and carbon emissions in China: A decomposition analysis," *Environ. Policy Governance*, vol. 25, no. 5, pp. 316–329, Sep. 2015.
[38] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 1901.
[39] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 7, pp. 498–520, 1933.
[40] J. Lin and D. A. Niemeier, "An exploratory analysis comparing a stochastic driving cycle to California's regulatory cycle," *Atmos. Environ.*, vol. 36, no. 38, pp. 5759–5770, Dec. 2002.



WEI YAN received the B.S. and M.S. degrees in power machinery and engineering from Shandong University, China, in 1994 and 1997, respectively, and the Ph.D. degree in control theory and engineering from Tongji University, China, in 2005.

He currently serves as the Combustion Branch Committee Member of Chinese Society for Internal Combustion Engines, the Associate Director of Shandong Intelligent Manufacturing Technology Application Center and the Project Evaluation Expert of Ministry of Education and Ministry of Finance. He currently takes charge of many projects, such as the National Key R&D Program, the 863 High-Tech Project, the National Science and Technology Support Program, and the National Natural Science Fund, Ministry's science and technology major projects, and so on. He has published more than 40 articles, including more than 30 articles indexed by SCI/EI. He also has obtained more than 20 patents and gotten six software copyrights. His research interests include big data analysis, cloud computing, and industrial internet application in manufacturing enterprises.

MEI-JING LI received the B.S. degree from the School of Energy and Power Engineering, Shandong University, China, in 2017, where she is currently pursuing the M.S. degree. Her research interests include driving cycle development, intelligent algorithm, and thermal management control systems.

YONG-CHANG ZHONG received the B.S. and M.S. degrees from the School of Energy and Power Engineering, Shandong University, China, in 2016 and 2019, respectively. His research interests include driving cycle development, intelligent algorithm, and control strategy of hybrid electric vehicles.

CHUN-YAN QU received the B.S. degree from the School of Energy and Power Engineering, Shandong University, in 2018, where she is currently pursuing the M.S. degree. Her research interests include control strategy of hybrid electric vehicles and transient process of diesel engines.

GUO-XIANG LI received the B.S. degree in vehicle engineering from Tsinghua University, China, in 1986, the M.S. degree in power engineering from Shandong University, China, in 1989, and the Ph.D. degree in power engineering from Shanghai Jiao Tong University, China, in 1996.

He currently serves as a Standing Director of the China Internal Combustion and the Chairman of the Basic Components Branch Engine Society, the Vice-Chairman and a Secretary-General of the Shandong Internal Combustion Engine Society, and so on. He currently engages in the research of emission control and reliability improvement technology of internal combustion engines and technology of hybrid vehicles. He has led the team to undertake eight Sub-projects of 863 National Projects, two projects of National Science and Technology Support Plan, five projects of National Key Research and Development Plan, and one project of National Natural Fund. He has won two special prizes for the progress of Science and Technology in China's Machinery Industry, published more than 80 articles indexed by SCI and EI, and authorized 12 national invention patents.