Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 3240675, 11 pages
https://doi.org/10.1155/2020/3240675

Research Article

Research on Phase Combination and Signal Timing Based on Improved K-Medoids Algorithm for Intersection Signal Control

Guojiang Shen, Xiangyu Zhu, Wei Xu, Longfeng Tang, and Xiangjie Kong

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China

Correspondence should be addressed to Xiangjie Kong; xjkong@ieee.org

Received 11 February 2020; Accepted 17 April 2020; Published 9 May 2020

Academic Editor: Bingxian Lu

Aiming at the problem of intersection signal control, a method of traffic phase combination and signal timing optimization based on the improved K-medoids algorithm is proposed. Firstly, the improvement of the traditional K-medoids algorithm embodies in two aspects, namely, the selection of the initial medoids and the parameter k, which will be applied to the cluster analysis of historical saturation data. The algorithm determines the initial medoids based on a set of probabilities calculated from the distance and determines the number of clusters k based on an exponential function, weight adjustment, and elbow ideas. Secondly, a phase combination model is established based on the saturation and green split data, and the signal timing is optimized through a bilevel programming model. Finally, the algorithm is evaluated over a certain intersection in Hangzhou, and results show that this algorithm can reduce the average vehicle delay and queue length and improve the traffic capacity of the intersection in the peak hour.

1. Introduction

With the rapid development of urban construction and socioeconomy, traffic congestion, one of China's urban diseases, not only brings tremendous pressure to urban traffic management but also seriously affects the harmonious development of cities. Many modern transportation facilities and applications can benefit from better performance of signal timing schemes [1–4]. For example, space-time road resources can be allocated more reasonably, the accuracy of traffic speed prediction can be improved [2], and the optimized signal cycle time and green split scheme can help make better-coordinated control [4]. In [5, 6], the authors studied the application of mobile crowdsourcing (MCS) in smart cities. In [7, 8], the authors integrate geographic and temporal influences into points of interest (POI) recommendations to help people find points of interest.

In recent years, several algorithms have been presented in the literature for traffic signal phase combination and timing optimization. In [9], the authors studied the dynamic prediction traffic signal control framework for a single intersection and optimized the signal timing according to the predicted arrival flow. In [10], a queuing and dissipation model of the intersection traffic flow was presented, which provided a theoretical basis for optimizing the intersection phase and timing. In [11], the authors considered an adaptive traffic signal control method based on fuzzy logic. This method optimized the phase duration and phase sequence. The results showed that the average queue length, the maximum queue length, and the parking rate were significantly shortened, but only lower queue lengths were considered. In [12], fog calculation was used to process traffic data, and a phase combination method based on a genetic algorithm was presented. The authors in [13] studied dynamic programming algorithms to optimize signal timing and phase, thereby, reducing average vehicle latency. In [14], the Artificial Bee Colony algorithm was adopted to optimize the signal cycle time and the green split, reducing the average vehicle delay and the average queue length, but the algorithm needed to obtain the vehicle speed online and calculate it. In [15], the authors considered a dynamic phase control method based on traffic flow, but it needed real-time detection and calculation of road conditions, resulting in poor practical application effect. In [16], the clustering algorithm was applied to process vehicle motion information, which was the basis for subsequent optimization, but only optimized the signal timing, excluding phase combination. In [17], a traffic signal segmentation algorithm based on the two-dimensional clustering was presented. It matched the best timing scheme for the current traffic conditions through the clustering analysis. However, the intersection traffic flow model cannot distinguish between a left turn and straight vehicles.

In [18], the authors studied the interval data-based K-means clustering method, and the clustering results can accurately describe the trend of traffic state evolution at an urban intersection. In [19], the K-means clustering algorithm was used to group traffic flows and divide the traffic condition level and provides a theoretical basis for matching the most suitable traffic signal control scheme in different situations. In [20], the author studies a dynamic traffic control method that predicts congestion by the clustering thought. In [21], a traffic signal control method based on the K-means clustering algorithm was presented, and the number of clusters was defined as two. The authors in [22] studied the improved affinity propagation (AP) clustering algorithm, which provided efficient and accurate traffic state information for traffic signal control. The average waiting time was effectively reduced. In [23], the authors studied the K-means clustering method to optimize the best switching time of time-of-day (TOD) control scheme, but the number of clusters needed to be specified in advance, which largely affected the effectiveness of the method. Similarly, the authors in [24] used the Kohonen cluster and K-means cluster to optimize TOD breakpoints and proved that K-means had a better performance. However, it was still necessary to specify the number of clusters and the initial cluster centers in advance, which was easy to fall into local optimum.

The existing researches mainly have the following shortcomings:

(1) the intersection traffic flow model is established without considering all of the flow directions

(2) the practical value of online data acquisition and frequent signal switching solutions is not high

(3) the number of clusters depends heavily on prior or empirical knowledge

To solve the problems above, this paper proposes a traffic phase combination and signal timing optimization method based on the improved K-medoids algorithm. Firstly, the improved K-medoids algorithm is used to cluster the historical saturation data, which can select the number of schemes k more quickly and accurately. Then, the phase combination model is established since K-medoids correspond to k pairs of saturation and green split data, which can combine the flow direction with similar traffic demand to improve the utilization of green time. Finally, the bilevel programming model is used to optimize the signal cycle time and green split of each phase, so that the timing scheme can be further optimized based on the phase combination. After clustering, each medoid composing a scheme library corresponds to a traffic scheme. In experiments, we choose an appropriate traffic scheme according to the Euclidean distance between the actual traffic saturation and medoids.

The paper is organized as follows: Section 2 introduces the traditional K-medoids clustering algorithm and its improvement. Section 3 designs the phase combination and signal timing optimization algorithm. Section 4 provides experimental results and comparisons with the traditional K-medoids algorithm. Section 5 provides conclusions and describes directions for future research.

2. Improved K-Medoids Algorithm

In this section, we first introduce the traditional k-medoids algorithm, then, to find better initial medoids and the appropriate parameter k, an improvement is introduced. Finally, we apply the improved k-medoids algorithm to the traffic saturation dataset into k clusters, and each cluster corresponds to one set of traffic scheme.

2.1. Traditional K-Medoids Algorithm. Clustering is an unsupervised learning algorithm that partitions the origin data into several clusters, where the data in the same cluster are similar to each other but different from the data in other clusters. K-medoids algorithm is a partition-based clustering algorithm. Compared with K-means clustering, it is less sensitive to outliers. Among many k-medoids algorithms, partitioning around medoids (PAM) is one of the most classical and powerful [25].

K-medoids algorithm first randomly selects k representative data points as the initial medoids, each medoid corresponds to one cluster. Secondly, Euclidean distance is applied to calculate the distance between all data and the chosen medoid, each data point will be assigned to the most similar medoid. Thirdly, such a new medoid in each cluster is found to minimize the criterion function within the cluster. The algorithm will stop until all of the medoids are equal to the previous ones, otherwise, assign each data to the nearest medoid and generate k new clusters. The Euclidean distance $d_{(x_i,y_i)}$ is used to measure the similarity between all of the data points and the medoids, which can be calculated as follows:

$$d_{(x_i,y_j)} = \sqrt{\left(x_{i1} - y_{j1}\right)^2 + \left(x_{i2} - y_{j2}\right)^2 + \cdots + \left(x_{in} - y_{jn}\right)^2}, \tag{1}$$

where x_i and y_i are both n-dimensional data objects.

The criterion function in within-cluster can be calculated as:

$$E_i = \sum_{b_j \in B_i} d_{\left(b_j,c_i\right)}{}^2, \tag{2}$$

where B_i is the cluster after clustering, b_j is the data point in the cluster B_i, and c_i is the medoid of the cluster B_i.

The criterion function is described as follows:

$$E = \sum_{i=1}^{k} \sum_{b_j \in B_i} {d_{(b_j,c_i)}}^2. \quad (3)$$

where k is the number of clusters.

2.2. The Improvement of K-Medoids Algorithm. For the K-medoids clustering algorithm, the number of clusters and the initialization have a great influence on the clustering process and results. In [26], a density peak clustering algorithm is proposed. This algorithm can select medoids and confirm the correct number of clusters. In [27], the author studied the K-medoids clustering algorithm based on a subset of candidate medoids and gradually increasing the number of clusters, thereby, improving the clustering performance of the algorithm. In order to reduce the negative impact when the initial medoids have a low dispersion degree, this paper proposes an initial point probability selection method based on the Euclidean distance. In addition, in order to reduce the artificial dependence for selecting initial medoids and avoid the excessive gap between each cluster, this paper proposes an optimization for selecting an optimal parameter k based on exponential function, weight adjustment, and elbow idea.

2.2.1. Improved Method for Selecting Initial Medoids. After selecting a point in sample data as the first medoid c_1 randomly, the Euclidean distance $d_{(b_h,c_i)}$ is applied to calculate the distance between each point b_h and the nearest medoid c_i, and the probability p_h that point b_h will be selected as the next cluster medoid can be calculated as:

$$p_h = \frac{{d_{(b_h,c_i)}}^2}{\sum_{b_j \in B} {d_{(b_j,c_i)}}^2}, \quad (4)$$

where B is the dataset, and the probability set P can be obtained as follows:

$$\mathrm{P} = [0, p_1, p_1 + p_2, \cdots, p_1 + p_2 + \cdots + p_{n-1}, 1], \quad (5)$$

where n is the number of samples in the dataset.

The roulette wheel method is used to select the cluster medoid $c_i (i \geq 2)$ (see Figure 1):

Step 1. We generate a random number r between $[0, 1)$, if r belongs to the interval $[p_1 + p_2 + \cdots + p_{i-1}, p_1 + p_2 + \cdots + p_{i-1} + p_i)$ in P, point b_i will be the second cluster medoid c_2.

Step 2. We recalculate the probabilities that each point in the dataset will be selected as the next medoid.

Step 3. We select the next medoid according to the probability set P and the roulette wheel method.

The steps mentioned above will be repeated until k centers are selected. The purpose is to make the initial medoids more discrete, which are closer to the real cluster centers.

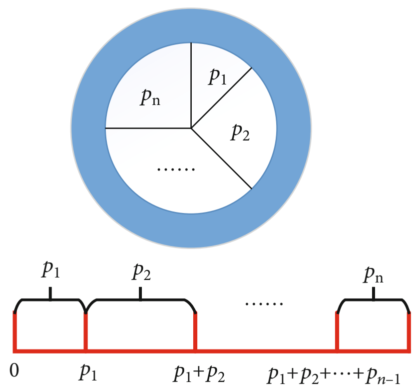

Figure 1: Roulette wheel method.

The number of iterations can be reduced, but settle the problem of trapping in a local optimum.

2.2.2. Improved Method for Selecting the Number of Clusters. The traditional criterion function in each cluster is the sum of all data within the cluster, which will make a big difference among clusters, and the classification will also be uneven. To settle the problem, this paper uses the exponential function e^x to optimize the criterion function calculation method. The criterion function in within-cluster can be calculated as:

$$S_i = e^{\sum_{b_j \in B_i} {d_{(b_j,c_i)}}^2} \quad (6)$$

In order to avoid exponential explosion, the weight coefficient t is employed, and the criterion function S can be calculated as follows:

$$S = \sum_{i=1}^{k} e^{\frac{\sum_{b_j \in B_i} {d_{(b_j,c_i)}}^2}{e}}. \quad (7)$$

With the optimization, the criterion function S can be calculated for different k. Following the increasement of parameter k, S will decrease. According to the elbow idea, S drops dramatically at the beginning, then, S reaches an elbow, finally, the curve of S turns to a plateau. The value k corresponding to the elbow is regarded as the optimal number of clusters.

2.3. Clustering with Saturation Data. Traffic saturation data is a collection of saturation at intersections, a single piece of data can be described as:

$$\mathrm{X}_i = [x_{i1}, x_{i2}, \cdots, x_{in}], \quad (8)$$

where n is the number of intersections.

The improved K-medoids algorithm described in Section 2.2 is then applied to the traffic saturation data, which divides the data into k clusters, and the initial cluster medoids are selected according to the distance probability p_h. The phase and timing optimization can be performed according to the cluster medoids, and each cluster corresponds to one set of traffic scheme, which means there will be k sets of initial traffic schemes.

3. Phase Combination and Signal Timing

In order to improve the adaptability of the traffic schemes for maching different traffic conditions, we establish the phase combination model and optimize the signal timing using the bilevel programming model.

3.1. Phase Combination Model. Signal phase refers to one or more flow directions displayed by the same signal lamp in a signal cycle time. The phase combination model mainly analyzes the conflicts of traffic flows with different directions, and use clustering ideas to merge nonconflicting flows with similar traffic characteristics into one phase. A reasonable combination can effectively reduce the release time of the green light, improve the utilization of the green light, and ensure transportation safety.

Two traffic flows are conflicting if there is a collision point of the vehicle travel path in these two directions. For example, the traffic flow in the east-west direction and the south-north direction are conflict, while the traffic flow in the east-west direction and the west-east direction are compatible. The conflict matrix can be constructed as follows:

$$\mathbf{C} = \begin{bmatrix} 0 & \varphi_{12} & \varphi_{13} & \cdots & \varphi_{1n} \\ \varphi_{21} & 0 & \varphi_{23} & \cdots & \varphi_{2n} \\ \varphi_{31} & \varphi_{32} & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \varphi_{(n-1)n} \\ \varphi_{n1} & \varphi_{n2} & \cdots & \varphi_{n(n-1)} & 0 \end{bmatrix}, \quad (9)$$

where φ_{ij} indicates whether the flow direction and j is conflict. If not, the value is 0, otherwise, 1.

The distance matrix is used to represent the difference between traffic flows, which can be constructed based on the saturation of flow directions, green signal split data, and the conflict matrix:

$$\mathbf{D} = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & 0 & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & d_{(n-1)n} \\ d_{n1} & d_{n2} & \cdots & d_{n(n-1)} & 0 \end{bmatrix}, \quad (10)$$

where the element d_{ij} in the matrix can be calculated as follows:

$$d_{ij} = \left(y_i - y_j\right)^2 + \varphi_{ij} = \left(x_i\lambda_i - x_j\lambda_j\right)^2 + \varphi_{ij}, \quad (11)$$

where y_i is the traffic flow ratio of the flow direction i, which reflecting the traffic demand not affected by the signal control scheme. x_i is the saturation of the flow direction i, and λ_i is the initial green split of the flow direction i.

Input:
The distance matrix $\mathbf{D}$
Output:
The final phase combination scheme D_{final}
Begin
1. for $i = 1$ to n, do
2, For $j = 1$ to $(i-1)$, do
3. If $d_{ij} < 1$ Then
4. $D_{first}.\mathrm{add}(d_{ij})$;
5. End If
6. End For
7. End For
8. While $c_1, c_2, c_3, \cdots, c_n! =$ each other, do
9. $D_{each} \longleftarrow [d_{c_1c_2}, d_{c_3c_4}, d_{c_5c_6}, \cdots, d_{c_{n-1}c_n}]$;
10. $D_{all}.\mathrm{add}(D_{each})$;
11. $S_{each} \longleftarrow \mathrm{sum}(D_{each})$;
12. $S_{all}.\mathrm{add}(S_{each})$;
13. End While
14. $S_{\min} \longleftarrow \mathrm{Selectmin}(S_{all})$;
15. $z \longleftarrow$ The index of $S_{\min}$ in S_{all} ;
16. $D_{final} \longleftarrow D_{all}[z]$;
End

Algorithm 1: Phase combination.

Since the distance between the flow direction i and j is the same as the distance between the flow direction j and i, the distance matrix is symmetric, that is, $d_{ij} = d_{ji}$. To ensure the balance of traffic flows in each phase, we optimize the phase combination according to the distance matrix between flow directions to make the combination more rational. For a typical crossroad, four-phase schemes are usually used, each phase consists of two flow directions, and the same flow direction traffic must be released only once in one cycle. Considering the symmetry of the distance matrix and all-zero values on the main diagonal, only the lower triangle needs to be processed. Algorithm 1 shows the optimization of the phase combination. If the distance between two flows is equal to or greater than 1, these two flows are physically conflicting. Hence, we select all the flow pairs with their distances less than 1 to form the D_{first} vector. If one scheme in the D_{first} contains all flow direction and each direction c_i only appears once, it will be saved as D_{each} to D_{all}. Then, we calculate the sum of the distances in D_{each} and insert it into the S_{all} as S_{each}, and the index z of the minimum $S_{\min}$ in S_{all} is selected. Finally, we choose the optimal scheme D_{final} according to z in D_{all}.

For example, there are two schemes here (see Figure 2): Scheme A takes east left movement and east through movement as one phase, west left movement and west through movement as another phase. Scheme B takes east left movement and west left movement as one phase, east through movement and west through movement as another phase, the distances of above four combinations are 0.2, 0.1, 0.3, and 0.4, respectively. The scheme A is chosen because the sum of the first two values is smaller than that of the last two values.

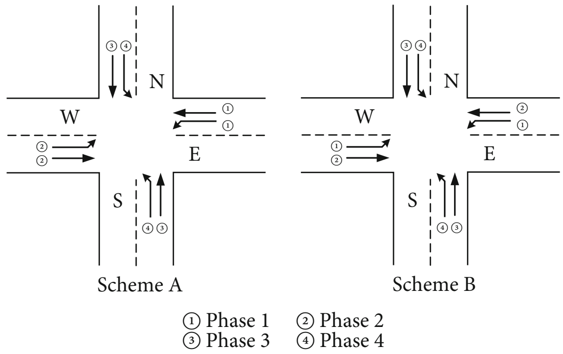

FIGURE 2: Schematic diagram of phase scheme comparison.

3.2. Traffic Signal Timing Bi-Level Programming Model. The bilevel programming model is a system optimization model with a two-tier hierarchical structure. The upper and lower levels have their own objective functions and constraints [28, 29]. The objective functions and constraints of the upper-level problem are not only related to the upper decision variables but also depending on the optimal solution of the lower level problem, while the optimal solution of the lower level problem is affected by the upper decision variables.

We establish a traffic signal timing optimization algorithm based on the bilevel programming model. The framework of the traffic signal timing optimization algorithm is shown in Figure 3.

3.2.1. Establishment of the Bilevel Programming Model. The signal cycle time is the key control parameter that determines the quality of traffic signal control in traffic signal timing, and the saturation can reflect the rationality of the signal cycle time to some extent. We establish the upper-level programming model with saturation as the decision target, which can be calculated as:

$$J = \min\left(\left|\bar{x} - \tilde{x}\right|\right), \tag{12}$$

where $\bar{x}$ is the average saturation of each phase, and $\tilde{x}$ is the target average saturation.

Under the condition of fixed signal cycle time constraints, the mean square error (MSE) of the saturation is used to evaluate the rationality of green split distribution. With the MSE, the lower-level programming model can be established as:

$$\sigma = \min\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2}\right), \tag{13}$$

where N is the number of signal phases. The saturation of each phase can be calculated as:

$$x_i = \frac{f_i}{q_i \lambda_i}, \tag{14}$$

where f_i is the arrival traffic flow for phase i, q_i is the average of each flow direction saturated flow in phase i, and λ_i is the initial green split of phase i.

3.2.2. Solution of the Bilevel Programming Model. The single-step action set with signal cycle time changes is designed to obtain the optimal signal cycle time of the upper-level programming model, the action set can be expressed as follows:

$$\boldsymbol{action}1 = \left[a_1, -a_1, 0\right], \tag{15}$$

where a_1, in seconds, is the adjustment step size for cycle time.

The three elements in action1 represent three operations, including addition, subtraction and invariance, respectively. For example, if the initial signal cycle time is T, the action1 is $[a_1, -a_1, 0]$, and the signal cycle time after each adjustment according to action1 will be $[T + a_1, T - a_1, T]$.

Algorithm 2 shows the optimization for signal cycle time. Each action of Equation (15) is executed in the initial signal cycle time T_0, and J is calculated by Equation (12) and (14), which will be inserted into J_{all}. The minimum $J_{\min}$ in J_{all} is selected, if its corresponding action is nonzero, the action will be taken, and the signal cycle time after execution will be updated as the initial scheme T_0 for the next iteration. The algorithm will loop until the action corresponding to $J_{\min}$ is zero, and the signal cycle time at this time $\mathbf{Now}[w]$ is regarded as the optimal signal cycle time T_f.

Similar to the upper level, to solve the optimal green split of the lower-level programming model, we design a set of single-step changes in the green time of each signal phase, the action set is

$$\begin{aligned} \text{action2} = {} & [[a_2, a_2, -a_2, -a_2], [a_2, -a_2, a_2, -a_2], [a_2, -a_2, -a_2, a_2], \\ & [-a_2, a_2, a_2, -a_2], [-a_2, a_2, -a_2, a_2], [-a_2, -a_2, a_2, a_2], \\ & [a_2, -a_2, 0, 0], [a_2, 0, -a_2, 0], [a_2, 0, 0, -a_2], \\ & [-a_2, a_2, 0, 0], [0, a_2, -a_2, 0], [0, a_2, 0, -a_2], \\ & [-a_2, 0, a_2, 0], [0, -a_2, a_2, 0], [0, 0, a_2, -a_2], \\ & [-a_2, 0, 0, a_2], [0, -a_2, 0, a_2], [0, 0, -a_2, a_2], [0, 0, 0, 0]], \end{aligned} \tag{16}$$

where a_2, in seconds, is the adjustment step size for green time.

The four elements in action2 represent the adjustment of green time of each phase in the four-phase scheme. For example, if the initial green time is $[g_1, g_2, g_3, g_4]$, the action2 is $[[a_2, a_2, -a_2, -a_2], \cdots, [0, 0, 0, 0]]$ and the green time of each phase after each adjustment according to action2 will be $[[g_1 + a_2, g_2 + a_2, g_3 - a_2, g_4 - a_2], \cdots, [g_1, g_2, g_3, g_4]]$.

Algorithm 3 shows the process for green split optimization. Considered the premise of the green split optimization algorithm that the signal cycle time is fixed, the sum of all elements in the action matrix is zero. According to the initial scheme of green split, the initial timing scheme is obtained by multiplying the signal cycle time. Each action of Equation (16) is executed, respectively, and then σ value of the corresponding action can be saved into $\boldsymbol{\sigma}_{\text{all}}$ according to Equation

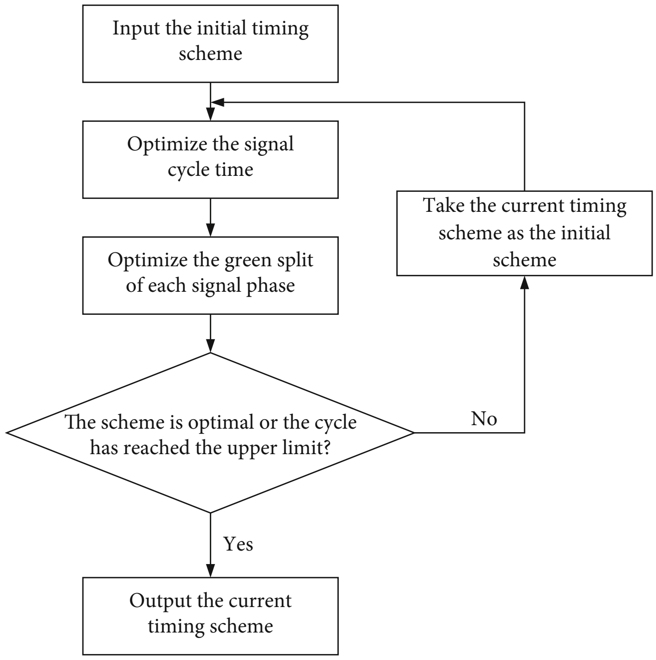

Figure 3: The framework of the traffic signal timing optimization algorithm.

Input:
The initial signal cycle time T_0, the average arrival traffic flow for each phase $\mathbf{f} = [f_1, f_2, \cdots, f_n]$, the average of each flow direction saturated flow in each phase $\mathbf{q} = [q_1, q_2, \cdots, q_n]$, the initial green split of each phase $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \cdots, \lambda_n]$ and the action set action1
Output:
The optimal signal cycle time T_f

```
Begin
1. w_0 ⟵ The index of 0 in action1 ;
2. w ⟵ A number! = w_0 ;
3. While w! = w_0, do
4.      For each a_i in action1, do
5.          TNow[i] ⟵ Calculate Adjust_T(T_0, a_i) ;
6.      End For
7.      J_all ⟵ [];
8.      For each TN_i in TNow, do
9.          J_all.add(Calculate J(TN_i, f_i, q_i, λ_i)) ;
10.     End For
11.     J_min ⟵ Selectmin(J_all) ;
12.     w ⟵ The index of J_min in J_all ;
13.     If w = = w_0 Then
14.         T_f ⟵ TNow[w] ;
15.     Else
16.         T_0 ⟵ TNow[w] ;
17.     End If
18. End While
End
```

Algorithm 2: Signal cycle time optimization.

(13) and (14). Then we select the minimum $\sigma_{\min}$ in $\boldsymbol{\sigma}_{\text{all}}$, if its corresponding action is not $[0, 0, 0, 0]$, the action will be taken, and the green timing scheme after execution is updated as the initial scheme $\boldsymbol{g}_0$ for the next iteration. The algorithm will loop until the action corresponding to $\sigma_{\min}$ is $[0, 0, 0, 0]$, and the green time of each signal phase at this time is converted into green split, and the optimal green split scheme λ_f is output.

We complete the green split optimization in the lower-level programming model, which will be fed back to the upper level. While in the upper level, the signal cycle time is optimized heuristically and iteratively under the restriction

Input:
The signal cycle time T, the arrival traffic flow for each phase $\mathbf{f} = [f_1, f_2, \cdots, f_n]$, the average of each flow direction saturated flow in each phase $\mathbf{q} = [q_1, q_2, \cdots, q_n]$, the initial green split of each phase $\boldsymbol{\lambda}_0 = [\lambda_{01}, \lambda_{02}, \cdots, \lambda_{0n}]$, and the action set action2
Output:
The optimal green split scheme$\boldsymbol{\lambda}_f = [\lambda_{f1}, \lambda_{f2}, \cdots, \lambda_{fn}]$

$I_0 \longleftarrow$ The index of $[0, 0, 0, 0]$ in action2 ;
$I \longleftarrow$ A number! $= I_0$;
For i in λ_0, do
  g_0.add($\lambda_{0i} T$) ;
While $I! = I_0$, do
  For each a_i in *action*2, do
    $GNow[i] \longleftarrow$ Calculate Adjust_g(g_0, a_i) ;
  End For
  $\sigma_{all} \longleftarrow []$;
  For each GN_i in $GNow$, do
    σ_{all}.add(Calculate $\sigma(GN_i, f_i, q_i)$) ;
  End For
  $\sigma_{\min} \longleftarrow$ Selectmin(σ_{all}) ;
  $I \longleftarrow$ The index of $\sigma_{\min}$ in σ_{all} ;
  If $I = = I_0$, do
    $g_f \longleftarrow GNow[I]$;
  Else
    $g_0 \longleftarrow GNow[I]$;
  End If
End While
For i in g_f, do
  λ_f.add(g_{fi}/T) ;
End

Algorithm 3: Green split optimization.

Figure 4: The actual layout of the intersection.

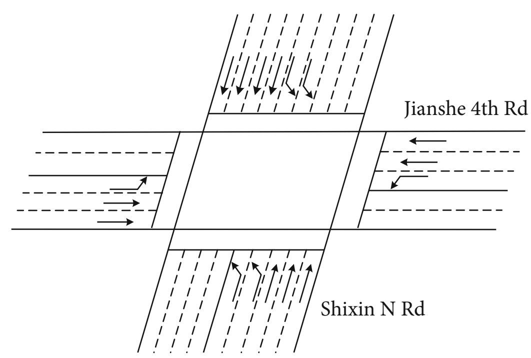

Figure 5: The simulation layout of the intersection.

of the green split, until the scheme is optimal or the cycle reaches the upper limit.

4. Simulation Experiment and Result Analysis

4.1. Experimental Methods and Experimental Data. The experiment is simulated in SUMO (Simulation of Urban Mobility), which is an open-source, highly portable, microscopic, and continuous traffic simulation software. The real-world intersection, Jianshe 4th Rd and Shixin N Rd in Xiaoshan District, Hangzhou, China (see Figure 4) is chosen as the operating environment.

According to the traffic laws and regulations in our country, the right turn movement can pass the intersection at any time without being controlled by the signal light; thus, only the left turn and the straight vehicles are considered in the simulation. Figure 5 shows the simulation structure of the intersection.

The traffic flow data were provided by the traffic control department of Xiaoshan District, Hangzhou, from 7 : 00 a.m. to 9 : 00 a.m. on November 20th, 2018. The original data was the traffic flow data of the signal cycle time and the timing scheme of the corresponding time, which was processed into saturation data set for clustering, and then, timing

Table 1: Traffic flow data of each flow direction in each time period of the intersection (veh/h).

Time	LE	SE	LS	SS	LW	SW	LN	SN
7:00	128	242	168	476	92	186	266	368
7:30	202	364	150	980	96	198	320	798
8:00	124	240	184	758	88	164	238	662
8:30	138	275	143	752	98	150	282	760
9:00	118	220	186	576	102	148	224	448
Saturation flow	1529	1641	1347	2360	1286	1606	1722	2228

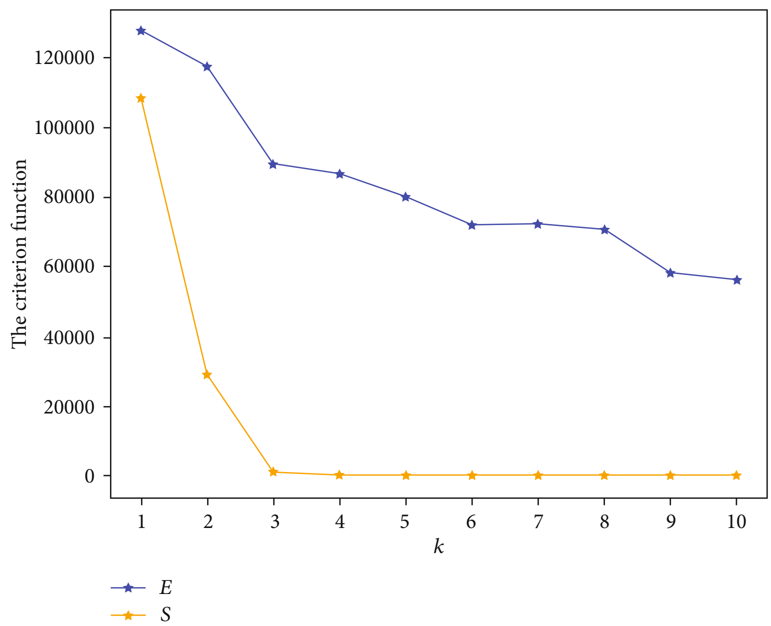

Figure 6: Algorithm comparison on the saturation dataset.

optimization was carried out based on the original timing scheme. Then, the data were divided into 5 periods, and flows of each direction were calculated every half an hour. The traffic flow data at a certain point in time is the average flow of the adjacent 15 minutes, that is, the traffic flow at 8:00 a.m. is the average traffic flow from 7:45 a.m. to 8:15 a.m. The flow data were divided into eight flow directions, such as the left turn for eastbound movement, the through for eastbound and so on. The average traffic flow of all lanes in each flow direction is recorded in Table 1, which has been converted into the hourly traffic flow to the inlet, and the through flow of each flow direction is also recorded. In this table, "E, " "S," "W," and "N" refer to eastbound, southbound, westbound, and northbound, respectively. "L" and "S" mean left turn and straight vehicles. For example, "LE" represents the traffic flow of the left turn in the eastbound movement.

The signal timing scheme generated by the improved K-medoids clustering algorithm is compared with the scheme generated by the traditional one to ensure the fairness of the experiment. In order to avoid the exponential explosion and make the criterion function E and S be in the same order of magnitude, the weight coefficient t is set as 11000. Additionally, we set the target average saturation δ_x to 70 according to the actual intersection traffic demand. In order to avoid missing the optimal timing scheme due to overlarge step size, the signal cycle time adjustment step a_1 and the green time adjustment step a_2 are both set as 1.

Table 2: Performance comparison of the traditional and improved K-medoids algorithm.

Algorithm	Number of clusters	Average execution time(s)
Traditional K-medoids	[3–6]	1.764
Improved K-medoids	3	1.623

In addition, the proposed algorithm is compared to the fixed phase scheme and the traffic flow and vector angle based on the optimization scheme [17].

4.2. Analysis of Results. The criterion functions of different k using traditional and improved K-medoids algorithm are shown, expressed by E and S, respectively. As k increases, the criterion functions decrease, and the rate of decline also stabilizes. In both cases, the optimal k is 3, while using the

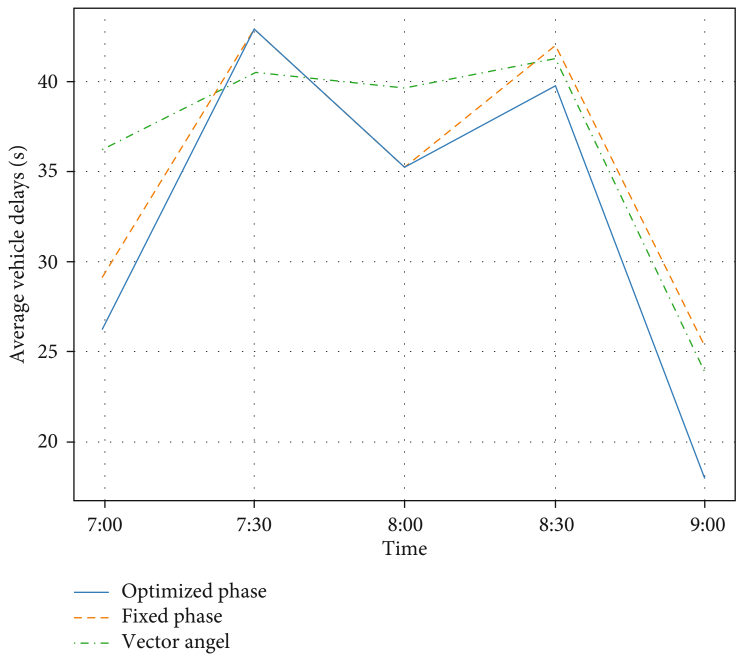

Figure 7: Comparison of average vehicle delays at intersection.

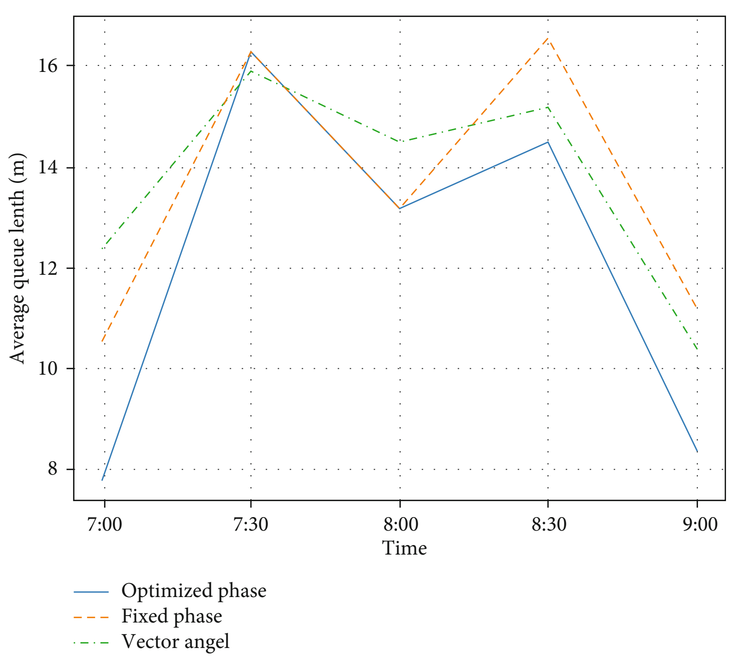

Figure 8: Comparison of average queue length at intersection.

improved K-medoids, it is easier to reach the result, and the elbow point can be identified more unambiguously. Figure 6 shows the curves of both algorithms, which is more intuitive.

Table 2 shows the different performances of the traditional and improved K-medoids algorithm. As for the number of clusters, in different iterations, the traditional K-medoids may reach the elbow when k is in range of 3 and 6, which is ambiguous to identify, while the improved K-medoids can

Table 3: Comparison of three optimization schemes at intersection.

	Average vehicle delays (s)	Average queue length (m)
Fixed phase	34.842	13.554
Vector angle	36.304	13.668
Optimized phase	32.380	12.012

Table 4: Delay comparison of three optimization schemes in flow direction.

Time	Fixed phase			Vector angle			Optimized phase		
	SS	SW	LE	SS	SW	LE	SS	SW	LE
7 : 00	14.44	23.89	35.57	20.66	45.82	37.62	15.91	28.88	27.28
7 : 30	29.81	24.04	40.20	21.35	53.38	41.17	29.82	24.05	40.19
8 : 00	22.74	23.61	32.67	29.84	55.28	44.68	22.73	23.62	32.67
8 : 30	35.35	20.92	39.16	27.55	55.99	50.72	29.18	26.45	30.31
9 : 00	12.21	25.20	34.72	11.79	31.85	29.78	14.92	30.91	21.41
Average	22.91	23.53	36.46	22.24	48.46	40.79	22.51	26.78	30.37

always reach the elbow when $k = 3$. In addition, the improved K-medoids runs faster than the traditional version, and that may because we optimize the selection of initial optimizing, which reduces the number of interactions.

Average vehicle delay and average queue length are used to evaluate the performance of the proposed algorithm.

Figures 7 and 8 show the curves of optimized phase and timing schemes under different conditions compared to fixed schemes that optimize only timing and vector angle-based schemes. The outperformance of our proposed method can be seen in all time periods. Table 3 shows the averaged values of the above two evaluation indexes, we can see that the proposed method outperforms the fixed phase method with improvements of 2.462 s (7.07%), and 1.542 m (11.38%) on the vehicle delay and the queue length, and also shows improvements of 3.924 s (10.81%) and 1.656 m (12.16%) compared to the traffic flow and vector angle-based optimization scheme.

Table 4 shows the delay comparison of three optimization schemes in SS, SW, and LE. We can see that the method proposed by us has a great improvement on the average vehicle delays in each flow direction compared to the traffic flow and vector angle-based optimization scheme. In our proposed method, the average vehicle delays of SW and LE is different from that of the fixed phase method, this is because the phase of SW and LE has changed. Compared with the fixed phase method, the average vehicle delays of LE in our method are reduced, but the average vehicle delays of SW are increased. The main reason is that our method improves the overall traffic capacity of the intersection rather than the single flow direction.

5. Conclusions

In this paper, we optimize the traditional K-medoids clustering algorithm in terms of the clustering number and initial medoids selection. In order to match the changes of traffic flow in different time periods adaptively, the phase combination optimization model is established to optimize the phase, and the bilevel programming model aims to optimize the signal timing, which can maximize the utilization of green time. The proposed algorithm is optimized for each flow direction. Whereas the flow saturation may be different when the overall situation is similar, we will study the difference of different flow saturation to achieve the optimal control effect of the intersection.

Conflicts of Interest

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

Acknowledgments

The authors express their sincere thanks to the researchers of Institute of Computer Intelligent System, Zhejiang University of Technology, China, for their help. This work was partially supported by the National Natural Science Foundation of China under Grant No. 61701443.

References

[1] G. Shen, L. Zhu, J. Lou, S. Shen, Z. Liu, and L. Tang, "Infrared multi-pedestrian tracking in vertical view via Siamese convolution network," *IEEE Access*, vol. 7, pp. 42718–42725, 2019.

[2] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, "Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (May, 2018)," *IEEE Access*, vol. 6, pp. 51756–51765, 2018.

[3] G. Shen, X. Han, J. Zhou, Z. Ruan, and Q. Pan, "Research on intelligent analysis and depth fusion of multi-source traffic data," *IEEE Access*, vol. 6, pp. 59329–59335, 2018.

[4] G. Shen and Y. Yang, "A dynamic signal coordination control method for urban arterial roads and its application," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 9, pp. 907–918, 2016.

[5] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities : technologies , applications , and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019.

[6] X. Kong, F. Xia, J. Li, M. Hou, M. Li, and Y. Xiang, "A shared bus profiling scheme for smart cities based on heterogeneous mobile crowdsourced data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1436–1444, 2020.

[7] W. Wang, J. Chen, J. Wang, J. Chen, and Z. Gong, "Geography-aware inductive matrix completion for personalized point of interest recommendation in smart cities," *IEEE Internet of Things Journal*, 2019.

[8] W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, and Z. Gong, "Trust-enhanced collaborative filtering for personalized point of interests recommendation," *IEEE Transactions on Industrial Informatics*, 2019.

[9] Z. Yao, L. Shen, R. Liu, Y. Jiang, and X. Yang, "A dynamic predictive traffic signal control framework in a cross-sectional vehicle infrastructure integration environment," *IEEE Transactions*

on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1455–1466, 2020.

[10] S. Le Vine, X. Liu, F. Zheng, and J. Polak, "Automated cars: Queue discharge at signalized intersections with 'Assured-Clear-Distance-Ahead' driving strategies," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 35–54, 2016.

[11] A. Vogel, L. Oremovic, and R. Simic, "Fuzzy Traffic Light Control Based on Phase Urgency," in *2019 International Symposium ELMAR*, Zadar, Croatia, September 2019.

[12] C. Tang, S. Xia, C. Zhu, and X. Wei, "Phase timing optimization for smart traffic control based on fog computing," *IEEE Access*, vol. 7, no. 8, pp. 84217–84228, 2019.

[13] Y. Guo, J. Ma, C. Xiong, X. Li, F. Zhou, and W. Hao, "Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: combined dynamic programming and shooting heuristic approach," *Transportation Research Part C: Emerging Technologies*, vol. 98, no. 1, pp. 54–72, 2019.

[14] C. Ma, W. Hao, A. Wang, and H. Zhao, "Developing a coordinated signal control system for urban ring road under the vehicle-infrastructure connected environment," *IEEE Access*, vol. 6, no. 5, pp. 52471–52478, 2018.

[15] Y. Wang and G. Zheng, "Research on polling based traffic signal control strategy with fuzzy control," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, Chengdu, China, December 2018.

[16] H. Rashid, "Intelligent traffic light control based on clustering using vehicular ad-hoc networks," in *2015 7th Conference on Information and Knowledge Technology (IKT)*, pp. 1–6, Urmia, Iran, May 2015.

[17] W. Hao, "Traffic signal segmentation algorithm based on two-dimensional clustering of traffic volume and vector angles," in *2017 International Smart Cities Conference (ISC2)*, Wuxi, China, September 2017.

[18] W. Rao, J. Xia, W. Lyu, and Z. Lu, "Interval data-based k -means clustering method for traffic state identification at urban intersections," *IET Intelligent Transport Systems*, vol. 13, no. 7, pp. 1106–1115, 2019.

[19] H. Chu and C. Wang, "Using K-means algorithm for the road junction time period analysis," in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, pp. 441–446, Taichung, Taiwan, November 2017.

[20] A. Rao, A. Phadnis, A. Patil, T. Rajput, and P. Futane, "Dynamic traffic system based on real time detection of traffic congestion," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–5, Pune, India, August 2018.

[21] J. Datesh, W. T. Scherer, and B. L. Smith, "Using k-means clustering to improve traffic signal efficacy in an IntelliDrive^{SM} environment," in *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*, pp. 122–127, Vienna, Austria, June-July 2011.

[22] W. Liu, G. Qin, Y. He, and F. Jiang, "Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8667–8681, 2017.

[23] X. Wang, W. Cottrell, and S. Mu, "Using k-means clustering to identify time-of-day break points for traffic signal timing plans," in *Proceedings 2005 IEEE Intelligent Transportation Systems, 2005*, pp. 586–591, Vienna, Austria, September 2005.

[24] Y. Jun and Y. Yang, "Optimization of time of day break points of intersection using cluster method," *International Journal of Advancements in Computing Technology*, vol. 5, no. 6, pp. 479–487, 2013.

[25] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*, Academic Press, 2008.

[26] A. Rodriguez and A. Laio, "Machine learning. Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[27] D. Yu, G. Liu, M. Guo, and X. Liu, "An improved K-medoids algorithm based on step increasing and optimizing medoids," *Expert Systems with Applications*, vol. 92, pp. 464–473, 2018.

[28] X. Wang, Y. Chang, and P. Zhang, "Traffic signal optimization based on system equilibrium and bi-level multi-objective programming model," in *Green Intelligent Transportation Systems. GITSS 2016. Lecture Notes in Electrical Engineering, vol 419*, W. Wang, K. Bengler, and X. Jiang, Eds., pp. 429–438, Springer, Singapore, 2018.

[29] G. Zhang, J. Han, and J. Lu, "Fuzzy bi-level decision-making techniques : a survey," *International Journal of Computational Intelligence Systems*, vol. 9, pp. 25–34, 2016.