

Article

Emerging Data-Driven Calibration Research on an Improved Link Performance Function in an Urban Area

Ming Chen ¹, **Kai Huang** ², **Jian Wang** ^{3,*}, **Wenzhi Liu** ¹ and **Yuanyuan Shi** ¹

¹ Laboratory for Traffic & Transport Planning Digitalization, Transport Planning and Research Institute, Ministry of Transport China, Beijing 100028, China; mingchenseu@outlook.com (M.C.); liuwz@tpri.org.cn (W.L.); smhmagic_007@126.com (Y.S.)

² School of Instrument Science and Engineering, Wuxi Campus, Southeast University, Wuxi 214000, China; kaihuang@seu.edu.cn

³ School of Transportation, Southeast University, Nanjing 211189, China

* Correspondence: jianw@seu.edu.cn

Abstract: The reliability of urban transportation systems is crucial for ensuring smooth traffic flow and minimizing disruptions caused by external factors. This study focuses on improving the stability and efficiency of transportation systems through the calibration of a refined link performance function while building upon the U.S. Bureau of Public Roads (BPR) model. To achieve this, we propose three customized algorithms—Newton's method, Bayesian optimization, and the differential evolutionary algorithm—to calibrate the key parameters. Additionally, we conducted a sensitivity analysis to assess the influences of the model parameters on link performance. Numerical experiments conducted in Yuyao City demonstrate the applicability and efficacy of the proposed model and solution algorithms. Our results reveal that the Newton approach is notably more efficient than the Bayesian optimization algorithm and the differential evolutionary algorithm.

Keywords: parameter calibration; link performance function; BPR function; license plate recognition data; parallel computing



Citation: Chen, M.; Huang, K.; Wang, J.; Liu, W.; Shi, Y. Emerging Data-Driven Calibration Research on an Improved Link Performance Function in an Urban Area. *Appl. Sci.* **2023**, *13*, 13318. <https://doi.org/10.3390/app132413318>

Academic Editor: Jose Santamaria

Received: 15 October 2023

Revised: 30 November 2023

Accepted: 11 December 2023

Published: 17 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

An effective urban transportation system not only makes sure that individuals have access to high-quality transportation services but also fosters the development of the entire society and economy. As urbanization and the density of the urban population increase, transportation in cities faces a growing number of challenging problems. Due to the high traffic flow, city roads become disorganized and inefficient, which causes enormous economic losses for both individual users and society as a whole. As a result, the objective of current research—both local and foreign—is to lessen this long-standing problem of urban congestion.

Traffic flow is essentially the set of all travel users' path selection behaviors, and it corresponds to traffic assignment in traffic planning [1–4]. Within modern traffic assignment methodologies, constraints on road capacity that affect travel speed or travel time are characterized through the use of link performance functions [5,6]. These performance functions mathematically illustrate the relationship between traffic volume and travel time (or cost) on the road. The link performance function is an essential component of a transportation system, as it describes the relationship between the link travel time and loading for road transportation. The link performance function is used to determine the optimal routes for travelers and, in turn, the overall efficiency of the road system. Over the years, many scholars have worked on this topic with the aim of better understanding the mechanism of the performance function and improving its accuracy [7–12]. Link performance functions can be used to predict traffic flows on specific road segments or a road network. They help decision makers develop traffic policies to deal with differing traffic conditions. A link

performance function model is very important in road network analysis, and it serves as the foundation for a network's traffic distribution and the evaluation of the quality of network traffic. However, the complexity of traffic flow and the peculiarities of diverse surroundings bring a huge challenge. It is difficult to determine a trustworthy set of parameters for a link performance function with universal applicability [13]. Therefore, it is necessary to recalibrate the function according to the traffic flow data in different spatial and temporal environments [14–20]. Parameter calibration problems have also been extensively studied in the transportation field [14,21–30].

The general form of the link performance function is shown below.

This is a first example of such an equation:

$$T = f(T_0, v, c). \quad (1)$$

Here, T_0 denotes the link's free-flow journey time (when the traffic flow equals zero), v signifies the link's traffic volume, c denotes the link's capacity, and T denotes the estimated link travel time.

Several of the most commonly used link performance functions are shown in Table 1.

Table 1. Several link performance functions.

Model	Mathematical Formula
BPR [31]	$T(T_0, v, c) = T_0 \left[1 + \alpha \left(\frac{v}{c} \right)^\beta \right]$
Davidson [32]	$T(T_0, v, c) = T_0 \left[1 + \frac{\alpha v}{c - v} \right]$
Akcelik [33]	$T(T_0, v, c) = T_0 \left\{ 1 + 0.25 \frac{d}{T_0} \left[\frac{v}{c} - 1 + \sqrt{\frac{v}{c} - 1 + 8 \frac{J}{Q} \frac{v}{c}} \right] \right\}$

One of the earliest studies on link performance functions was performed by the U.S. Bureau of Public Roads (BPR), which is still widely used today [31]. The set of its parameters is $[\alpha, \beta] = [0.15, 4]$. Since then, many researchers have proposed modifications to the original formula, including additional variables of road conditions, traffic flow, and driver behavior. It has the advantage that the roadway flow is not limited by roadway capacity. It is unnecessary to detect feasible solutions when assigning traffic, thus making the assignment algorithm significantly simpler. This model was established based on the velocity–flow curve from the 1965 version of the Highway Capacity Manual (HCM). However, a limitation of this model is that it tends to produce high-velocity estimates when the saturation rate is greater than 1 and low estimates when the saturation rate is smaller than 1. With the introduction of the 1994 version of the HCM, the velocity–flow curve became flatter. To accommodate this change, the BPR function was modified, leading to the Metropolitan Transportation Commission (MTC) model [34]. The structure of the MTC model is similar to that of the BPR model, except that the parameter values have been adjusted. Additionally, a new BPR function model for urban trunk roads was proposed with $[\alpha, \beta] = [0.05, 10]$ as the set of model parameters [35]. According to Spiess [5], excessive values of β can result in numerical problems. These include overflow circumstances, loss of precision, and a low rate of convergence by unfairly penalizing overloaded links during the initial iterations of an equilibrium assignment. Additionally, a BPR function with high values of β always produces free-flow speeds; it does not correspond to the real traffic volumes for lines with volumes that are significantly below capacity.

Davidson [32] proposed a link performance function based on the queuing theory to improve the BPR model. It failed to adapt to actual traffic conditions when the flow was close to capacity. The Davidson function can be applied to any type of road infrastructure. However, the model has been the subject of considerable debate due to some contradictions in interpreting its parameters. As a result, several researchers proposed various improved models based on the Davidson function [33,36,37], among which the Akcelik model has gained extensive usage.

By using the coordinate transformation method, Akcelik [33] suggested a time-dependent version of the Davidson function that sought to account for the intersection delay. When intersection delays give a sizable amount of link travel speed, this delay function enhances the modeling of the link travel speed. A parameter is used to capture the delay. Higher values are preferred for arterial highways without coordinated signal systems, whereas lower values are recommended for motorways. When comparing other speed delay functions to data gathered from 119 freeway segments in California, it was found that the Akcelik function was frequently the best fit for planning applications [35]. According to Singh [38], applying the Akcelik function in traffic assignment also has some additional benefits, such as improved convergence and more accurate speed prediction in crowded areas. The compactness of the Akcelik function makes it beneficial because it allows for the universal use of a single functional form and the avoidance of complicated parameters while determining intersection delays.

This paper aims to report on data-driven calibration research for an improved link performance function. It provides a comprehensive overview of the calibration and validation of the link performance function and then provides a sensitivity analysis to evaluate the effect of different parameter levels on link performance. Three solution algorithms are proposed to cope with the calibration problem. In addition, a value of the time parameter is introduced and calibrated against income data to measure the value of time for different populations and trip types.

The rest of this paper is organized as follows. Section 2 describes the model in depth and examines the model parameters. The recommended solution approaches are presented in Section 3. The efficiency of the proposed methodology is validated in a genuine case study in Section 4. Finally, some concluding remarks are provided.

2. Model

2.1. Problem Description

In this paper, the calibration of a link performance function is studied. Congestion-related time consumption is a significant determinant of the route that a traveler chooses when using a road system. As a result, the connection between the journey time and loading for road transportation is a crucial component of the overall link performance function. Many academics have devoted their careers to related topics to clarify its function. Among these accomplishments, the BPR function is well known. The formula for it is as follows.

$$T_a(v_a) = T_a \left[1 + \alpha \left(\frac{v_a}{c_a} \right)^\beta \right]. \quad (2)$$

Here, T_a is the travel time under free-flow conditions on arc a , v_a represents the volume of arc a , and c_a denotes the capacity of arc a . α and β are regression parameters.

According to Equation (2), as the volume of arc a grows, so does the travel time. The travel time will dramatically rise if the volume exceeds the capacity. This could be an indication of the volume–delay relationship due to congestion.

Considering that the national conditions of China and the United States are different and that there are large differences in economic development and traffic conditions, the BPR function is not fully adapted to the road conditions in China. Therefore, the BPR function was improved and recalibrated in research on capacity in the National Key Transportation Technology Research and Development Program of China during the 9th Five-Year Plan Period, and it was combined with data from large-scale actual traffic surveys in China [39]. The form of this new model (for convenience, it is termed BPR95) is as follows.

$$T_a(v_a) = \frac{T_a}{\alpha_1} \left[1 + \left(\frac{v_a}{c_a} \right)^\beta \right], \quad (3)$$

$$\begin{cases} S_a = \frac{\alpha_1 S_d}{1 + \left(\frac{v_a}{c_a}\right)^\beta} \\ \beta = \alpha_2 + \alpha_3 \left(\frac{v_a}{c_a}\right)^3 \end{cases}. \quad (4)$$

Here, S_a is the average travel speed, S_d is the design speed, and $\alpha_1, \alpha_2, \alpha_3$ are parameters to be calibrated.

In addition to the time impedance, the monetary expenditure is also an important factor in the impediment of a road. Therefore, it is necessary to include money spent in the general cost function and to quantify time spent as money through the value of the time (VOT) parameter [40]. The general cost function for road transportation is shown as follows.

$$F_a(v_a) = r^{road} L_a + \theta T_a(v_a). \quad (5)$$

Here, $F_a(v_a)$ represents the general cost of arc a , r^{road} is the expenditure per kilometer on the road, L_a denotes the physical length of arc a , and θ is the VOT.

The link performance function model proposed in this paper consists of two major components: T_a , the time impedance, including the consumption of time and delays caused by tolls, road width, and other physical conditions on the section itself, as well as the characteristics of the traffic flow itself caused by time delays due to traffic congestion, is reflected in the BPR95 function; $r^{road} L_a$ for the cost impedance, including transportation costs and road tolls. The scale of the link performance function's value F_a is RMB (yuan). Therefore, the scale of θ is min/yuan and the scale of r^{road} is yuan/km.

The parameter calibration problem is formulated as an optimization problem, and it is used to minimize the error between the real travel time patterns (empirical data) and the model-based travel time patterns. The error measurement is defined as follows:

$$\min Z = \sqrt{\frac{\sum_i \left(\frac{T_i^{model} - T_i^{data}}{T_i^{data}} \right)^2}{N}}. \quad (6)$$

Here, T_i^{model} denotes the model-based link travel time, T_i^{data} represents the real link travel time, and N is the number of samples.

2.2. Model Analysis

Before calibrating the parameters, a sensitivity analysis was conducted for the link performance function to examine the effects of the parameter settings. In this instance, the Sobol sensitivity analysis method was applied [41,42]. Table 2 illustrates the results, where S_i stands for the first-order sensitivity index, and ST_i stands for the total sensitivity index. The portion of the output variance that is only attributable to changes in the input factor X_i is described by S_i . The total of S_i plus all X_i -related interaction effects is ST_i .

Table 2. Results of the sensitivity analysis.

	S_i	ST_i
α_1	1.697	1.008
α_2	5.5×10^{-5}	2.406×10^{-4}
α_3	2.00×10^{-6}	4.335×10^{-8}

α_1 has the biggest impact on the link performance, as seen in Table 2. α_1 is equivalent to a scaling factor for regulating the cost when expressed as a link performance function. In fact, α_2 and α_3 are utilized to regulate the curvature or growth trend of the link performance function, since they have relatively little impact on the cost and are both formally in the exponential section. We depict the link performance function for various parameter values,

as shown in Figure 1, where the free-flow time and α_1 are kept at 1, to see how different parameter sets affect the model.

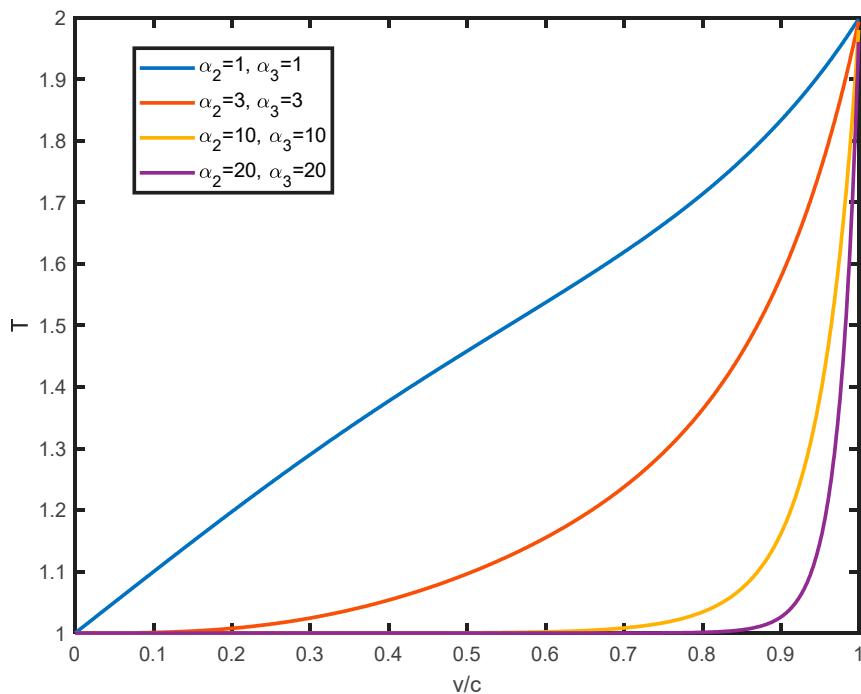


Figure 1. Curves of the link performance function.

The function index term grows as α_2 and α_3 increase, since v/c is less than or equal to 1. In this instance, the cost approaches the free-flow time when the saturation is low and develops more quickly when the saturation is close to 1, making the entire function's curve steeper where v/c tends to 1.

3. Solution Method

The BPR95 model is more complex in structure than the original BPR model and cannot be transformed into a linear model through appropriate transformations, making calibration more difficult. We propose three algorithms to solve this optimization problem.

3.1. Newton's Method

Newton's method [43], which is also known as the Newton–Raphson method, is a popular numerical optimization technique used to find the roots of a function. The method utilizes an iterative approach to progressively refine the approximation of the root. Starting with an initial guess for the root, the method uses the first and second derivatives of the function to calculate a new approximation of the root. This process is repeated until the approximation is sufficiently close to the true root. The strength of Newton's method lies in its fast convergence rate, which can be quadratic under certain conditions. However, the method is sensitive to the initial guess and may converge to a local minimum instead of the global minimum. Additionally, the method requires the computation of derivatives, which can be computationally expensive for complex functions.

The Gauss–Seidel method is used for iteration, and it has better convergence efficiency than that of the Jacobi method. The algorithm works as follows (Algorithm 1).

Algorithm 1 Newton's method for the calibration of the link performance function

```

1  Initialize:  $\alpha_1^0, \alpha_2^0, \alpha_3^0, j = 0$ 
2  Iterate:
3    Update variables sequentially:
4    Update  $\alpha_1^{i+1} = \alpha_1^i - \left( \frac{\partial Z(\alpha_1^i, \alpha_2^i, \alpha_3^i)}{\partial \alpha_1} \right) / \left( \frac{\partial^2 Z(\alpha_1^i, \alpha_2^i, \alpha_3^i)}{\partial \alpha_1^2} \right)$ 
5    Update  $\alpha_2^{i+1} = \alpha_2^i - \left( \frac{\partial Z(\alpha_1^{i+1}, \alpha_2^i, \alpha_3^i)}{\partial \alpha_2} \right) / \left( \frac{\partial^2 Z(\alpha_1^{i+1}, \alpha_2^i, \alpha_3^i)}{\partial \alpha_2^2} \right)$ 
6    Update  $\alpha_3^{i+1} = \alpha_3^i - \left( \frac{\partial Z(\alpha_1^{i+1}, \alpha_2^{i+1}, \alpha_3^i)}{\partial \alpha_3} \right) / \left( \frac{\partial^2 Z(\alpha_1^{i+1}, \alpha_2^{i+1}, \alpha_3^i)}{\partial \alpha_3^2} \right)$ 
7    Update  $j = j + 1$ 
8    Go back to Line 3 until the termination criterion is satisfied.

```

3.2. Bayesian Optimization

The calibration problem is a complex nonlinear optimization problem. However, Bayesian optimization (BO) is well suited to handle it. BO is a probabilistic model-based optimization algorithm that uses a surrogate model to approximate the objective function and an acquisition function to balance exploration and exploitation.

Studies have found that BO is widely applied in the solving of traffic optimization problems [44–47]. The BO framework consists of two main parts: using a probabilistic model to substitute the original evaluation of costly and intricate objective functions, such as the Gaussian process, and using the posterior information of the surrogate model to create active selection strategies, such as acquisition functions.

The Gaussian process is an infinite-dimensional generalization of the multivariate normal distribution. It is a convenient and robust prior distribution. Let (\mathbf{A}_i, Z_i) represent a sample in the calibration problem, where \mathbf{A}_i denotes the parameters in this problem and Z_i is the corresponding objective function value. Let $\vec{\mathbf{Z}} = [\dots, Z_i, \dots]^T$ denote the vector of function observations and let $\vec{\mu} = [\dots, E[Z_i], \dots]^T$ represent the vector of the expected values of function observations. The Gaussian process assumes that the vector of n samples $\{(\mathbf{A}_i, Z_i)\}_{i=1}^n$ generates a multivariate Gaussian distribution:

$$\vec{\mathbf{Z}} \sim \mathcal{N}(\vec{\mu}, \Sigma). \quad (7)$$

Here, Σ is the covariance matrix. The form of the distribution is decided by a mean function and a covariance function. In general, the expected value does not affect the optimization process, so a common practice is to set $E[Z_i]$ to a constant, i.e., $E[Z_i] = \mu_0(\mathbf{A}_i) = c$. In the Gaussian process, the covariance between Z_i and Z_j is measured by the distance between \mathbf{A}_i and \mathbf{A}_j , $\|\mathbf{A}_i - \mathbf{A}_j\|_2^2$. It is assumed that a smaller distance between \mathbf{A}_i and \mathbf{A}_j will lead to a higher correlation between Z_i and Z_j . The covariances are generally calculated using the kernel functions.

In this paper, the ARD Matérn 5/2 kernel is used:

$$\begin{aligned} \text{cov}(Z_i, Z_j) &= \kappa_{M52}(\mathbf{A}_i, \mathbf{A}_j) \\ &= \alpha \left(1 + \sqrt{5 \frac{\|\mathbf{A}_i, \mathbf{A}_j\|_2^2}{l^2} + \frac{5}{3} \frac{\|\mathbf{A}_i, \mathbf{A}_j\|_2^2}{l^2}} \right) \cdot \exp \left\{ -\sqrt{5 \frac{\|\mathbf{A}_i, \mathbf{A}_j\|_2^2}{l^2}} \right\}. \end{aligned} \quad (8)$$

Here, α and l are parameters of this kernel. This covariance function is twice differentiable. These parameters can be estimated by using the maximum likelihood estimation approach.

For any new sample (\mathbf{A}', Z') , the prior distribution can be rewritten as:

$$\begin{bmatrix} \vec{\mathbf{Z}} \\ Z' \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \vec{\mu} \\ \mu_0(\mathbf{A}') \end{bmatrix}, \begin{bmatrix} \Sigma & \text{cov}(\vec{\mathbf{Z}}, Z') \\ \text{cov}(\vec{\mathbf{Z}}, Z')^T & \text{cov}(Z', Z') \end{bmatrix} \right). \quad (9)$$

The predicted distribution or posterior probability distribution is given by:

$$Z' | \{(\mathbf{A}_i, Z_i)\}_{i=1}^n, \mathbf{A}' \sim N(\mu_n(\mathbf{A}'), \sigma_n^2(\mathbf{A}')), \quad (10)$$

$$\mu_n(\mathbf{A}') = \text{cov}(\vec{\mathbf{Z}}, Z')^T \Sigma^{-1} (\vec{\mathbf{Z}} - \vec{\mu}) + \mu_0(\mathbf{A}'), \quad (11)$$

$$\sigma_n^2(\mathbf{A}') = \text{cov}(Z', Z') - \text{cov}(\vec{\mathbf{Z}}, Z')^T \Sigma^{-1} \text{cov}(\vec{\mathbf{Z}}, Z'). \quad (12)$$

Here, $\mu_n(\mathbf{A}')$ denotes the estimated value of Z' given $\{(\mathbf{A}_i, Z_i)\}_{i=1}^n$ and \mathbf{A}' , and $\sigma_n^2(\mathbf{A}')$ represents the variance of the estimation error introduced into the function observation.

Based on the posterior information, the acquisition function can determine the evaluation direction. The acquisition function is formulated considering both the estimated objective function value (i.e., the expectation in Equation (12)) and the uncertainty information (i.e., the variance in Equation (13)) over all of the feasible regions. Let af represent the acquisition function; then, the determination problem can be expressed as:

$$\mathbf{A}_{next} = \underset{\mathbf{A}}{\operatorname{argmax}} af(\mathbf{A}). \quad (13)$$

The design of the acquisition function is a significant part of BO. In this paper, the expected improvement (EI) function is applied.

Let Z_{best}^+ denote the current best value of the objective function. EI maximizes the expected improvement over Z_{best}^+ . It has a closed-form expression:

$$af_{EI}(\mathbf{A}) = (\mu(\mathbf{A}) - Z_{best}^+) \Phi \left(\frac{\mu(\mathbf{A}) - Z_{best}^+}{\sigma(\mathbf{A})} \right) + \sigma(\mathbf{A}) \phi \left(\frac{\mu(\mathbf{A}) - Z_{best}^+}{\sigma(\mathbf{A})} \right). \quad (14)$$

Here, Φ represents the cumulative distribution function of the standard normal distribution, and ϕ denotes the probability density function of the standard normal distribution. $\mu(\mathbf{A})$ and $\sigma(\mathbf{A})$ can be computed with Equations (12) and (13), respectively.

Compared with the original objective function, the acquisition function has relatively simple forms and is easier to optimize.

The algorithm of BO for solving the calibration of the link performance function is summarized as follows (Algorithm 2).

Algorithm 2 Bayesian optimization for the calibration of the link performance function

-
- Step 0:** Set $n = 1$ and $\max_n = N$. Randomly sample n parameter sets $\mathbf{T}_n = \{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n\}$. For each parameter set $\mathbf{A}_i, i \in \{1, 2, \dots, n\}$, calculate the objective function value $\mathbf{Z}_n = \{Z_1, Z_2, \dots, Z_n\}$.
- Step 1:** Build the Gaussian process regression model based on \mathbf{T}_n and \mathbf{Z}_n . Obtain the posterior distribution of Z_{n+1} based on the sampled points.
- Step 2:** Use the posterior information obtained in the previous step, and maximize the acquisition function to determine a new sample point \mathbf{A}_{n+1} .
- Step 3:** With \mathbf{A}_{n+1} , calculate the corresponding objective function value Z_{n+1} . Set $n = n + 1$ and update \mathbf{T}_n and \mathbf{Z}_n .
- Step 4:** Check the termination criterion. If $n = N$, return A_{best} , or return to Step 1.
-

3.3. Differential Evolution Algorithm

Heuristic algorithms are effective methods for solving parameter calibration problems. Heuristic search algorithms do not require information such as continuous and differentiable objective functions and have good global optimization capabilities, making them a hot topic in the field of optimization [48–50].

In this study, the differential evolution algorithm (DE) [51] was used for parameter calibration. The differential evolution algorithm is a heuristic stochastic search algorithm based on population differences. The problem-solving process is represented as the survival of the fittest of “chromosomes”. Through generations of evolution, including operations such as replication, crossover, and mutation of the “chromosome” population, it ultimately converges to the individual that is most adaptable to the environment, thereby obtaining the optimal or satisfactory solution to the problem. The algorithm was originally proposed for solving Chebyshev polynomials and is a post-set heuristic algorithm used for optimization problems. Essentially, it is a greedy genetic algorithm based on real-number encoding focusing on preserving the optimal solution.

DE is similar to the genetic algorithm in that it involves mutation, crossover operations, and elimination mechanisms. However, the main difference between DE and genetic algorithms is that the mutation part involves selecting the difference between two solution members, which is then scaled and added to the variables of the current solution member. Therefore, DE does not need to use probability distributions to generate the next generation of solution members.

Consider the following unconstrained optimization problem:

$$\min f(x_1, x_2, \dots, x_D), \quad (15)$$

$$\text{s.t. } x_j^L \leq x_j \leq x_j^U. \quad (16)$$

The DE algorithm consists of the following steps.

(1) Initialization of the population

The classical differential evolution algorithm uses real-valued encoding, which makes the algorithm more suitable for solving real-valued optimization problems.

$$\left\{ X_i(0) \mid x_{i,j}^L \leq x_{i,j}(0) \leq x_{i,j}^U; i = 1, 2, \dots, NP; j = 1, 2, \dots, D \right\}. \quad (17)$$

Here, $X_i(0)$ denotes the i -th individual, D denotes the dimensionality of the problem, and NP notes the population size.

Solution vectors are randomly generated within the upper and lower bounds:

$$x_{i,j}(0) = x_{i,j}^L + \text{rand}(0, 1)(x_{i,j}^U - x_{i,j}^L). \quad (18)$$

(2) Mutation

Individual mutation is achieved through a differential strategy. The common differential strategy is to randomly select two different individuals in the population, scale their vector difference, and combine it with the individual to be mutated.

$$V_i(g+1) = X_{r1}(g) + F(X_{r2}(g) - X_{r3}(g)). \quad (19)$$

Here, r_1, r_2, r_3 are three random numbers in the range of $[1, NP]$. F is the scaling factor, which is also known as the mutation rate, and g represents the g -th generation.

Generally, the mutation operator often takes a constant value, which is difficult to determine accurately. If the mutation rate is too large, the global optimal solution is low, and if the mutation rate is low, the diversity of the population will decrease, so the phenomenon of “premature convergence” is likely to occur. Therefore, we can design an adaptive mutation operator:

$$\lambda = e^{1 - \frac{G_m}{C_m + 1 - G}}, F = F_0 \times 2^\lambda. \quad (20)$$

In this way, the mutation operator starts with $2F_0$, and it can maintain diversity in the early stages and prevent premature convergence. As the evolution progresses, the mutation operator decreases until it becomes F_0 , avoiding the destruction of the optimal solution.

The differential evolution algorithm is named after the differential mutation strategy, which is the most important part of the algorithm.

(3) Crossover

The purpose of the crossover operation is to randomly select individuals, as differential evolution is also a random algorithm. The differential evolution algorithm uses discrete crossover factors, including binomial crossover and exponential crossover. The crossover operator discretely crosses the mutation vector V_i generated by the mutation operator with the parent individual vector X_i to obtain the trial vector U_i . The method of the crossover operation is as follows:

$$U_{i,j}(g+1) = \begin{cases} V_{i,j}(g+1) & \text{if } \text{rand}(0, 1) \leq CR \\ x_{i,j}(g) & \text{otherwise} \end{cases}. \quad (21)$$

Here, CR is called the crossover probability, and it has a range of $[0, 1]$. It is used to generate new individuals with a certain probability.

(4) Selection

The selection step in the differential evolution algorithm uses a greedy strategy, where the fitter individuals are chosen as new individuals after the offspring population is generated through the mutation and crossover operators. This is done using a binary tournament selection operator, where each offspring competes with its corresponding parent, and the one that is more fit is selected to be included in the next generation's population. For minimization problems, the selection operator can be described as follows:

$$X_i(g+1) = \begin{cases} U_i(g+1) & \text{if } f(U_i(g+1)) \leq f(X_i(g)) \\ X_i(g) & \text{otherwise} \end{cases}. \quad (22)$$

Because the differential evolution algorithm uses one-to-one tournament selection, it is a steady-state evolution algorithm that preserves elite individuals. Once a new population is formed, the differential evolution algorithm continues to evolve the population through mutation, crossover, and selection operators until the termination condition is met, and the program is exited.

The algorithm of DE for solving the calibration of the link performance function is summarized as follows (Algorithm 3).

Algorithm 3 DE algorithm for the calibration of the link performance function

```

1 Input: DE control parameters (mutation probability; crossover probability), the fit-
ness function (the objective function of the calibration problem)
2 Randomly initialize the population
3 Main Loop:
4   Mutation
5   Crossover
6   Selection
7   Repeat the main loop until the termination criterion is satisfied.

```

4. Case Study**4.1. Study Area and LPR Data Analysis**

Parameter calibration belongs to the research on the micro-characteristics of traffic flow, which requires high data accuracy. Therefore, the accuracy of traffic survey data significantly impacts the results. Based on survey experience, on-site manual surveys often result in large errors and many uncertainties, and the reproducibility of the data is weak, which can easily lead to the failure of the final calibration results. Therefore, more accurate survey methods should be used for traffic flow characteristic surveys.

This section includes a case study to demonstrate the model and solution suggested in this paper. The study was conducted in the central area of Yuyao City, China, as shown on the map in Figure 2. The study area covered an area of approximately 40 m² and consisted of 31 LPR detectors. The data analyzed in this case study included almost 80,000 records from June to July 2018. Table 3 illustrates the LPR data used in the study. The case study was performed using Matlab 2021b running on a desktop with an Intel(R) Core (TM) i5-11400F CPU @ 2.60 Hz, 2.59 GHz, and 16.00 G RAM.

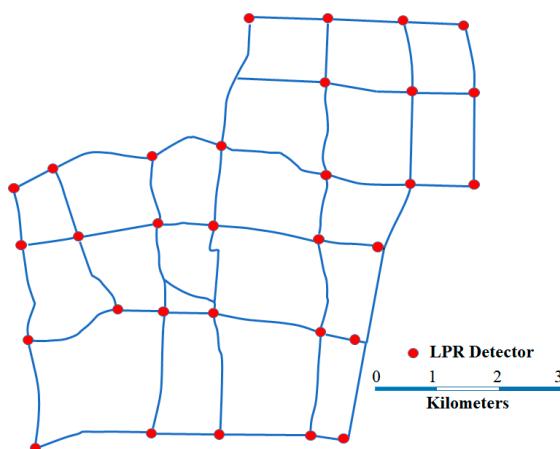


Figure 2. Map of the study area.

Table 3. Example of the LPR data.

License Plate No.	Timestamp	Intersection No.	Intersection Name	Lane No.	Speed	Entrance Direction
ZB 1234	15 June 2018 07:00:00	330281000000011086	Intersection of Chengdong Road and Ziling Road	02	40	02

Notes: The license plate number begins with ZB (the county code), then has four capital letters or digits; the intersection number and name specifically match the LPR detector. The lane numbers start with "01" from the outside to the inside; the speed is expressed in kilometers per hour; with reference to the entrance direction, "01" denotes north, "02" denotes east, "03" denotes south, and "04" denotes west.

The LPR data were used to calculate the traffic flows, which were calculated by aggregating the same origins and destinations within the study area.

4.2. Results

4.2.1. Calibration of the Link Performance Function

In this study, we evaluated the performance of Newton's method, Bayesian optimization (BO), and differential evolution (DE) when calibrating the optimal set of parameters of a link performance function. The hyperparameters of DE were determined according to a predefined guideline. The convergence criteria for all algorithms were set to $Z < 10^{-10}$. The optimal set of parameters was $[\alpha_1, \alpha_2, \alpha_3] = [0.93, 1.8799, 4.8507]$.

The results suggest that, in terms of solution efficiency, Newton's approach performed better than BO and DE, as seen in Figure 3. This was because, while DE and BO involve several calculations of the objective function values during the solution process, which can take a significant amount of time, Newton's approach is a gradient-based method with local second-order convergence. It was also shown how much of an advantage BO had over DE. The potential to achieve better solutions in fewer function evaluations was what separated BO from DE. This was because BO could take advantage of the information from each evaluation of the objective function value.

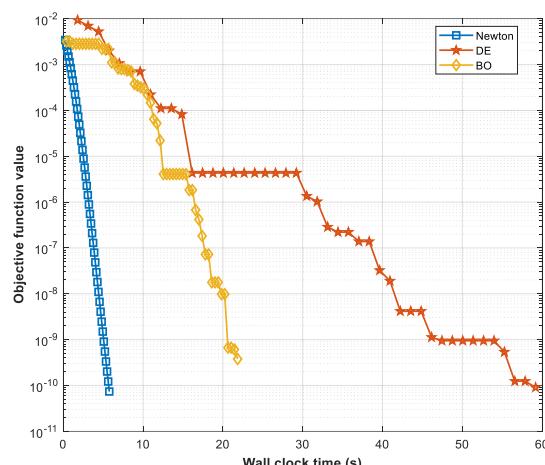


Figure 3. Results of calibration.

In recent decades, the performance of computers has experienced exponential growth due to the rapid development of semiconductor technology. To improve computational efficiency, parallel computing techniques have become a major approach. In this context, the DE algorithm is suitable for designing a parallel computing framework due to its individuals being independent of each other, with no coupling between them. As a result, the update and selection of individuals can be conducted in parallel. The results in Figure 4 include a comparison of the solution efficiency of Newton's method, BO, DE, and DE_par (a parallel version of DE). The experiments were conducted using a six-core CPU to evaluate the performance of these methods.

To measure the acceleration effect, the absolute acceleration ratio was calculated according to the following equation:

$$r = \frac{t_s}{t_p}. \quad (23)$$

Here, t_s is the serial execution time, and t_p is the parallel execution time.

The DE algorithm's serial execution time was 59.173 s, while its parallel execution time was 20.398 s, resulting in a speedup ratio of 2.9 (<6). The parallelization of DE significantly improved its performance. Although it remained slightly inferior to Newton's method, it performed a little better than BO. As a global optimization algorithm, DE can always converge to the global optimum, which is its strength over Newton's method.

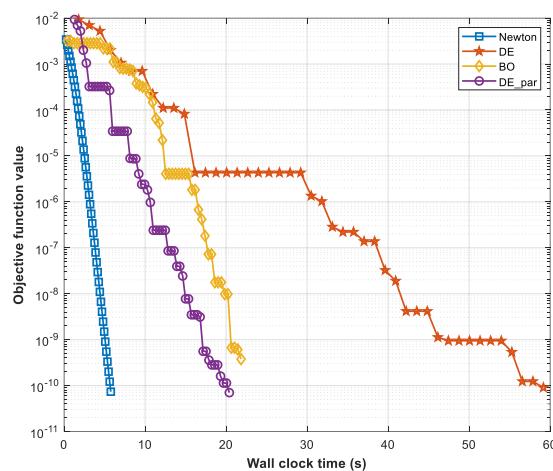


Figure 4. Results of calibration (with DE_par algorithm added).

4.2.2. Calibration of the VOT

(a) VOT of travel for work

From a macroeconomic perspective, it is considered that labor is one of the factors of production, and the savings in travel time spent by travelers can be used in national production, which is equivalent to an increase in the factors of production and, thus, leads to an increase in GDP. Based on this approach, the formula for calculating the VOT is

$$VOT = GDP / (P \times T). \quad (24)$$

Here, GDP is the gross national product, P is the average annual employment, and T is the average annual number of hours worked by an individual.

The city's gross regional product per capita based on the household population was CNY 172,912 (equivalent to USD 26,802 at the average annual exchange rate).

The hourly wage rate per capita is

$$\begin{aligned} VOT_{work} &= 172912 / 250 / 8 \\ &= 86.456. \end{aligned} \quad (25)$$

Existing studies have shown that this production method is applicable for reflecting the time value of the flow of the group of passengers who travels with working time (or productive time), i.e., it is more applicable to work trips. At the same time, travelers should theoretically be regular occupants.

(b) VOT of travel for non-work

For non-work travel by fixed and non-fixed occupants, the income method was used to measure their VOT. The income method reflects the opportunity cost of the time taken by the trip relative to the individual, i.e., the reduction in income caused by the time being consumed by the travel process and resulting in the traveler not being able to work. The calculation formula is

$$VOT = INC / T. \quad (26)$$

Here, INC denotes the annual personal income in RMB (yuan).

In contrast to the production method, the income method is applicable for reflecting the VOT of the flow of a group of passengers who pay for trips with personal income, and this is more suitable for non-work (or non-production) trips.

As there is a strong correlation between the VOT and income, the monthly income was divided into intervals, and the correspondence between the time value and individual monthly income was obtained, as shown in Table 4.

Table 4. The VOT of travel for non-work.

Monthly income range (yuan)	(0, 3000]	(3000, 6000]	(6000, 10,000]
VOT range (yuan/hour)	(0, 17.05]	(17.05, 34.09]	(34.09, 56.82]
Monthly income range (yuan)	(10,000, 15,000]	(15,000, 20,000]	(20,000, $+\infty$]
VOT range (yuan/hour)	(56.82, 85.23]	(85.23, 113.64]	(113.64, $+\infty$]

5. Conclusions

Traffic assignment computations heavily rely on the link performance function. The ultimate computation results are significantly influenced by the parameters chosen. It is difficult to obtain the right values based only on experience because the physical importance of these characteristics is not always obvious. As a result, this research suggested a technique for investigating the calibration of link performance function parameters that are appropriate for the unique features of big cities. Numerous major cities' active traffic planning projects can greatly benefit from this information.

In this study, the parameter calibration problem was solved using Newton's technique, BO, and the DE algorithm. The results showed that Newton's method was substantially more effective than DE and BO. Nonetheless, the DE approach is naturally parallelizable, and we greatly sped it up by parallelizing the work on a six-core CPU. Furthermore, we used a variety of methods to classify journeys into business and non-work travel categories in order to calculate the value of travel (VOT).

Future studies should focus on using cutting-edge technologies, such as machine learning and deep learning, to assess the performance and impedance of roadways in order to increase the precision of path planning [52].

Author Contributions: Conceptualization, M.C. and K.H.; methodology, M.C. and J.W.; software, M.C.; validation, M.C. and J.W.; formal analysis, M.C. and W.L.; resources, Y.S. and W.L.; writing—original draft preparation, M.C.; writing—review and editing, J.W., K.H. and W.L.; visualization, M.C. and K.H.; supervision, Y.S.; project administration, J.W.; funding acquisition, Y.S., J.W. and K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Laboratory for Traffic and Transport Planning Digitalization Program (No. (2022)JH-F05), the Key Laboratory Open Fund of the Transportation Industry in Comprehensive Transportation Theory (MTF2023001) and the National Natural Science Foundation of China Youth Program (No. 72301066).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Qiu, J.; Huang, K.; Hawkins, J. The taxi sharing practices: Matching, routing and pricing methods. *Multimodal Transp.* **2022**, *1*, 100003. [[CrossRef](#)]
- Zhou, G.; Huang, K.; Mao, L. Design of Commute Carpooling Based on Fixed Time and Routes. *Int. J. Veh. Technol.* **2014**, *2014*, 634926. [[CrossRef](#)]
- Zhou, G.; Lv, M.; Bao, T.; Mao, L.; Huang, K. Design of intelligent carpooling program based on big data analysis and multi-information perception. *Clust. Comput.* **2019**, *22*, 521–532. [[CrossRef](#)]
- Huang, K.; Liu, F.; Hu, Y.; Liu, Z. An analysis of the taxi-sharing organizing and pricing. In *Green Intelligent Transportation Systems: Proceedings of the 7th International Conference on Green Intelligent Transportation System and Safety 7*; Springer: Berlin, Germany, 2018; pp. 263–276.
- Spiess, H. Conical volume-delay functions. *Transp. Sci.* **1990**, *24*, 153–158. [[CrossRef](#)]
- Branston, D. Link capacity functions: A review. *Transp. Res.* **1976**, *10*, 223–236. [[CrossRef](#)]
- Huntsinger, L.F.; Roushail, N.M. Bottleneck and queuing analysis: Calibrating volume–delay functions of travel demand models. *Transp. Res. Rec.* **2011**, *2255*, 117–124. [[CrossRef](#)]

8. Nie, X.; Zhang, H. Delay-function-based link models: Their properties and computational issues. *Transp. Res. Part B Methodol.* **2005**, *39*, 729–751. [[CrossRef](#)]
9. Kucharski, R.R.; Drabicki, A. Estimating Macroscopic Volume Delay Functions with the Traffic Density Derived from Measured Speeds and Flows. *J. Adv. Transp.* **2017**, *2017*, 4629792. [[CrossRef](#)]
10. Russo, F.; Vitetta, A. Reverse assignment: Calibrating link cost functions and updating demand from traffic counts and time measurements. *Inverse Probl. Sci. Eng.* **2011**, *19*, 921–950. [[CrossRef](#)]
11. Yang, H.; Xu, W.; He, B.-S.; Meng, Q. Road pricing for congestion control with unknown demand and cost functions. *Transp. Res. Part C Emerg. Technol.* **2010**, *18*, 157–175. [[CrossRef](#)]
12. Neuhold, R.; Fellendorf, M. Volume Delay Functions Based on Stochastic Capacity. *Transp. Res. Rec. J. Transp. Res. Board* **2014**, *2421*, 93–102. [[CrossRef](#)]
13. Foytik, P.; Cetin, M.; Robinson, R.M. Calibration of BPR function based on link counts and its sensitivity to varying demand. In Proceedings of the Transportation Research Board 92nd Annual Meeting, Washington, DC, USA, 13–17 January 2013.
14. Cheng, Q.; Liu, Z.; Lin, Y.; Zhou, X. An s-shaped three-parameter (S3) traffic stream model with consistent car following relationship. *Transp. Res. Part B Methodol.* **2021**, *153*, 246–271. [[CrossRef](#)]
15. Lu, Z.; Meng, Q.; Gomes, G. Estimating link travel time functions for heterogeneous traffic flows on freeways. *J. Adv. Transp.* **2016**, *50*, 1683–1698. [[CrossRef](#)]
16. Hou, G.; Chen, S.; Bao, Y. Development of travel time functions for disrupted urban arterials with microscopic traffic simulation. *Phys. A Stat. Mech. Its Appl.* **2022**, *593*, 126961. [[CrossRef](#)]
17. Wollenstein-Betech, S.; Sun, C.; Zhang, J.; Cassandras, C.G.; Paschalidis, I.C. Joint Data-Driven Estimation of Origin–Destination Demand and Travel Latency Functions in Multiclass Transportation Networks. *IEEE Trans. Control Netw. Syst.* **2022**, *9*, 1576–1588. [[CrossRef](#)]
18. Balakrishna, R.; Ben-Akiva, M.; Koutsopoulos, H.N. Offline calibration of dynamic traffic assignment: Simultaneous de-mand-and-supply estimation. *Transp. Res. Rec.* **2007**, *2003*, 50–58. [[CrossRef](#)]
19. Ran, B.; Rouphail, N.M.; Tarko, A.; E Boyce, D. Toward a class of link travel time functions for dynamic assignment models on signalized networks. *Transp. Res. Part B Methodol.* **1997**, *31*, 277–290. [[CrossRef](#)]
20. Kachroo, P.; Sastry, S. Traffic Assignment Using a Density-Based Travel-Time Function for Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1438–1447. [[CrossRef](#)]
21. Cheng, Q.; Liu, Z.; Guo, J.; Wu, X.; Pendyala, R.; Belezamo, B.; Zhou, X. Estimating key traffic state parameters through parsimonious spatial queue models. *Transp. Res. Part C Emerg. Technol.* **2022**, *137*, 103596. [[CrossRef](#)]
22. Xing, J.; Liu, R.; Zhang, Y.; Choudhury, C.F.; Fu, X.; Cheng, Q. Urban network-wide traffic volume estimation under sparse deployment of detectors. *Transp. A Transp. Sci.* **2023**, *2197511*. [[CrossRef](#)]
23. Zhou, X.S.; Cheng, Q.; Wu, X.; Li, P.; Belezamo, B.; Lu, J.; Abbasi, M. A meso-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to volume-delay function with inflow demand-to-capacity ratio. *Multimodal Transp.* **2022**, *1*, 100017. [[CrossRef](#)]
24. Akuh, R.; Zhong, M.; Raza, A.; Dong, Y. A method for evaluating the balance of land use and multimodal transport system of new towns/cities using an integrated modeling framework. *Multimodal Transp.* **2023**, *2*, 100063. [[CrossRef](#)]
25. Almeida, L.S.; Goerlandt, F. An ant colony optimization approach to the multi-vehicle prize-collecting arc routing for connectivity problem. *Multimodal Transp.* **2022**, *1*, 100033. [[CrossRef](#)]
26. Parishwad, O.; Jiang, S.; Gao, K. Investigating machine learning for simulating urban transport patterns: A comparison with traditional macro-models. *Multimodal Transp.* **2023**, *2*, 100085. [[CrossRef](#)]
27. Xiao, D.; Kim, I.; Zheng, N. Recent advances in understanding the impact of built environment on traffic performance. *Multimodal Transp.* **2022**, *1*, 100034. [[CrossRef](#)]
28. Durrani, U.; Lee, C. Calibration and Validation of Psychophysical Car-Following Model Using Driver’s Action Points and Perception Thresholds. *J. Transp. Eng. Part A Syst.* **2019**, *145*, 04019039. [[CrossRef](#)]
29. Omrani, R.; Kattan, L. Concurrent Estimation of Origin–Destination Flows and Calibration of Microscopic Traffic Simulation Parameters in a High-Performance Computing Cluster. *J. Transp. Eng. Part A Syst.* **2018**, *144*, 04017068. [[CrossRef](#)]
30. Ghiasi, A.; Hussain, O.; Qian, Z.; Li, X. Lane Management with Variable Lane Width and Model Calibration for Connected Automated Vehicles. *J. Transp. Eng. Part A Syst.* **2020**, *146*, 04019075. [[CrossRef](#)]
31. *Highway Capacity Manual*; Transportation Research Board Special Report 209; Transportation Research Board, National Research Council: Washington, DC, USA, 1985.
32. Davidson, K.B. A flow travel time relationship for use in transportation planning. In Proceedings of the 3rd Australian Road Research Board (ARRB) Conference, Sydney, Australia, 5–9 September 1966.
33. Akçelik, R. Travel time functions for transport planning purposes: Davidson’s function, its time dependent form and alternative travel time function. *Aust. Road Res.* **1991**, *21*, 49–59.
34. Singh, R. *Updating Speed-Flow and Speed-Capacity Relationships in Traffic Assignment: 1990 Mtc Regional Travel Demand Model Development Project: Draft*; University of California: Berkeley, CA, USA, 1993.
35. Skabardonis, A.; Dowling, R. Improved Speed-Flow Relationships for Planning Applications. *Transp. Res. Rec. J. Transp. Res. Board* **1997**, *1572*, 18–23. [[CrossRef](#)]

36. Davidson, K.B. The theoretical basis of a flow-travel time relationship for use in transportation planning. *Aust. Road Res.* **1978**, *8*, 23–35.
37. Rose, G.; Taylor, M.A.P.; Tisato, P. Estimating travel time functions for urban roads: Options and issues. *Transp. Plan. Technol.* **1989**, *14*, 63–82. [[CrossRef](#)]
38. Singh, R.; Dowling, R. Improved speed-flow relationships: Application to transportation planning models. Commonwealth of Massachusetts, Executive Office of Transportation and Construction; and Boston Metropolitan Planning Organization. In Proceedings of the Seventh TRB Conference on the Application of Transportation Planning Methods Transportation Research Board, Boston, MA, USA, 7–11 March 2002.
39. Wang, Y.; Zhou, W.; Lyu, L. Research on the theory and application of road impedance function. *J. Highw. Transp. Res. Dev.* **2004**, *21*, 82–85.
40. Paletti, R.; Vovsha, P.; Givon, D.; Birotker, Y. Impact of individual daily travel pattern on value of time. *Transportation* **2015**, *42*, 1003–1017. [[CrossRef](#)]
41. Sobol, I.M. Sensitivity analysis for non-linear mathematical models. *Math. Model. Comput. Exp.* **1993**, *1*, 407–414.
42. Saltelli, A.; Ratto, M.; Andres, T.; Campolongo, F.; Cariboni, J.; Gatelli, D.; Saisana, M.; Tarantola, S. *Global Sensitivity Analysis. The Primer*; John Wiley & Sons, Ltd.: London, UK, 2007.
43. Dembo, R.S.; Eisenstat, S.C.; Steihaug, T. Inexact newton methods. *SIAM J. Numer. Anal.* **1982**, *19*, 400–408. [[CrossRef](#)]
44. Huo, J.; Liu, Z.; Chen, J.; Cheng, Q.; Meng, Q. Bayesian optimization for congestion pricing problems: A general framework and its instability. *Transp. Res. Part B Methodol.* **2023**, *169*, 1–28. [[CrossRef](#)]
45. Huo, J.; Liu, C.; Chen, J.; Meng, Q.; Wang, J.; Liu, Z. Simulation-based dynamic origin–destination matrix estimation on freeways: A Bayesian optimization approach. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *173*, 103108. [[CrossRef](#)]
46. Otsuka, T.; Shimizu, H.; Iwata, T.; Naya, F.; Sawada, H.; Ueda, N. Bayesian optimization for crowd traffic control using multi-agent simulation. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, Auckland, New Zealand, 27–30 October 2019; pp. 1981–1988.
47. Sha, D.; Ozbay, K.; Ding, Y. Applying Bayesian Optimization for Calibration of Transportation Simulation Models. *Transp. Res. Rec. J. Transp. Res. Board* **2020**, *2674*, 215–228. [[CrossRef](#)]
48. Umar, M.; Sabir, Z.; Raja, M.A.Z.; Baskonus, H.M.; Ali, M.R.; Shah, N.A. Heuristic computing with sequential quadratic programming for solving a nonlinear hepatitis B virus model. *Math. Comput. Simul.* **2023**, *212*, 234–248. [[CrossRef](#)]
49. Umar, M.; Sabir, Z.; Raja, M.A.Z.; Amin, F.; Saeed, T.; Sanchez, Y.G. Design of intelligent computing solver with Morlet wavelet neural networks for nonlinear predator–prey model. *Appl. Soft Comput.* **2023**, *134*, 109975. [[CrossRef](#)]
50. Gómez-Aguilar, J.F.; Sabir, Z.; Alqhtani, M.; Umar, M.; Saad, K.M. Neuro-Evolutionary Computing Paradigm for the SIR Model Based on Infection Spread and Treatment. *Neural Process. Lett.* **2022**, *55*, 4365–4396. [[CrossRef](#)]
51. Liu, J.; Lampinen, J. A fuzzy adaptive differential evolution algorithm. In Proceedings of the IEEE TENCOM’02, 2002 IEEE Region 10 Conference on Computer, Communications, Control and Power Engineering, Beijing, China, 28–31 October 2002.
52. Huo, J.; Wu, X.; Lyu, C.; Zhang, W.; Liu, Z. Quantify the Road Link Performance and Capacity Using Deep Learning Models. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18581–18591. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.