sustainability

MDPI

Article

A Hybrid Teaching-Learning-Based Optimization Algorithm for the Travel Route Optimization Problem alongside the Urban Railway Line

Fuying Liu [1,*], Chen Liu [2], Qi Zhao [1] and Chenhao He [1]

[1] JangHo Architecture College, Northeastern University, Shenyang 110819, China; dbdxzq@163.com (Q.Z.); haozi17614008817@163.com (C.H.)
[2] School of Resources and Civil Engineering, Northeastern University, Shenyang 110819, China; 18940106208@163.com
* Correspondence: liufuying@mail.neu.edu.cn

Abstract: Accurate travel route optimization is essential to promote and grow tourism in modern society. This paper investigates a travel route optimization problem alongside the urban railway line and proposes a hybrid teaching–learning-based optimization (HTLBO) algorithm. First, a mathematical programming model is established to minimize the total traveling time, in which the routes between and in different cities have to be appropriately determined. Then, a hybrid metaheuristic named HTLBO is proposed for solution generation. In HTLBO, depth first search (DFS) is utilized to obtain the optimal routes of any two stations in railway network, and a three-level coding method is designed to accommodate the problem characteristic. Besides, opposition-based learning (OBL) is embedded into teaching-learning-based optimization (TLBO) for enhancing HTLBO's exploration ability, while variable neighborhood descent (VND) is used to enhance the algorithm's exploitation ability. Finally, a case study is presented and simulation results verify HTLBO's feasibility and effectiveness.

Keywords: optimization; metaheuristic; teaching-learning-based optimization; depth first search; variable neighborhood descent

check for updates

Citation: Liu, F.; Liu, C.; Zhao, Q.; He, C. A Hybrid Teaching-Learning-Based Optimization Algorithm for the Travel Route Optimization Problem alongside the Urban Railway Line. *Sustainability* **2021**, *13*, 1408. https://doi.org/10.3390/su13031408

Received: 19 December 2020
Accepted: 25 January 2021
Published: 29 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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

With the rapid development of social economy, the people's living standards have been significantly improved in China. Tourism is now becoming an essential part of peoples' daily life [1,2]. In such contexts, scientific planning of travel routes is necessary to reduce traffic time and to improve tourist satisfaction [3]. This study is motivated by this background and aims to deal with a travel route optimization problem alongside the urban railway line.

As mentioned above, this study investigated an optimization problem in which the total traveling time has to be minimized. Theoretically, the considered problem can be regarded as a variant of the classic travel salesman problem (TSP) [4], and Figure 1 presents a schematic diagram. In the specific problem setting, a traveler is planning to visit all tourist attractions in a set of cities. The way he travels between different cities is via the railway, and the railway network associated with these cities is predefined. For the considered problem, decisions should be made are described as follows: (1) the visiting sequence of different cities, (2) the railway route between different cities, and (3) the traveling routes in each city.

For the considered problem, the inherent complexity of the combinatorial optimization problem and the coupling of different subproblems make it difficult to obtain a satisfactory solution [5]. Besides, the specific constraints associated with practical applications further increase its difficulty. Therefore, this research is motivated in this context and focuses on designing an efficient decision-making method for solution generation.

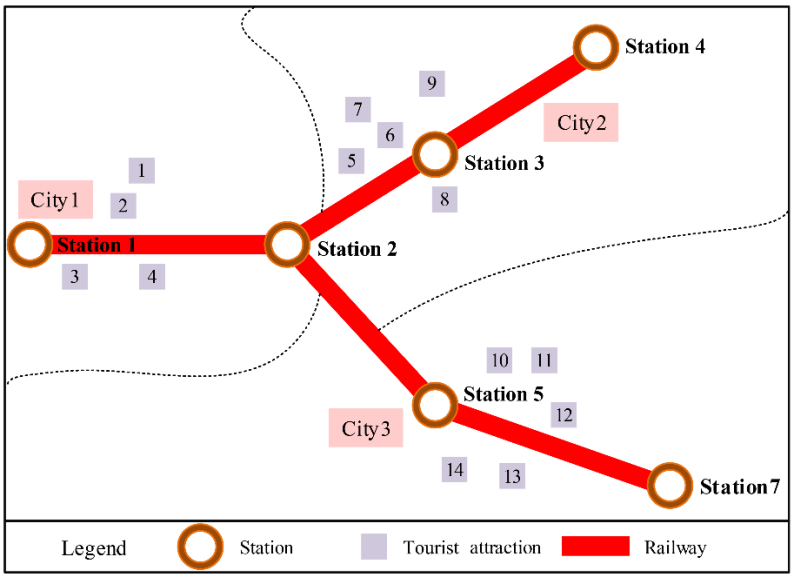

Figure 1. Schematic diagram of the considered problem.

A great many algorithms have been designed and introduced to solve travel route optimization problems, which can be divided into three categories: exact algorithms, heuristic methods, and metaheuristic algorithms [6]. Exact algorithms include dynamic programming, branch and bound algorithms, among others. These algorithms are capable of solve small-scale instances optimally. However, the computation time increases sharply with the increasing of instance size, which makes it impossible to solve medium and large-scale instance in reasonable time [7–9]. Heuristic methods were applied to solve different optimization problems by virtue of some classical scheduling rules, and the efficiency is impressive [10,11]. Despite that, such applications have strict requirements on the problem's structure setting, and the algorithms' generality are very poor. Under such circumstances, metaheuristic approaches are now becoming the most appropriate tools since these algorithms are problem-agnostic and can be adjusted to coordinate the problem-specific information [12]. Some effective metaheuristics have been designed and applied for solution generation of similar problems, including Genetic Algorithm (GA), Particle swarm optimization (PSO), and others. [13,14]. Meanwhile, there is a theorem called no free lunch (NFL) that has proven that there is no approach is best suited to tackle all optimization problems [15]. In other words, the superior performance of a metaheuristic approach on one problem cannot guarantee similar performance on other optimization problems [16]. This theorem is the foundation of many researches works that allows researchers in different fields to adjust different algorithms for their problems or propose some new algorithms. This is also the motivation and foundation of our research work, in which a novel metaheuristic method name HTLBO is designed and applied to solve the considered travel route optimization problem.

TLBO is one of the most recent metaheuristics, which has been proven to precede other classics metaheuristics in some applications [17]. It is inspired by behaviors of a classroom and contains two phases: teacher and learner phases. TLBO-related approaches with better algorithm performance have already been proposed by a great many researchers [18–22]. For most recent research work, please refer to the book presented by Rao [23]. Nevertheless, there still exist some challenges when introducing TLBO for different optimization problems [24]. First, TLBO is initially proposed for continuous problems and thus requires some modifications when applied to discrete problems. Second, it tends to get trapped into local optima when solving real-world scale instances, which is now a common weakness of population-based metaheuristics. Thus, researchers in different fields attempt to design appropriate coding and decoding schemes when introducing TLBO to discrete problems as well as to strengthen algorithm's exploitation ability. The current research does make

efforts in these regards and presents a new TLBO version for the considered travel route optimization problem.

To our best knowledge, the current research work is the first attempt to develop a TLBO-version algorithm for the travel route optimization problem alongside the urban railway line. Four main contributions of this research work are summarized as follows:

- A mathematical model for the considered travel route optimization problem is formulated with an objective of minimizing the total traveling time.
- The HTLBO algorithm is proposed to solve considered problem by combining TLBO metaheuristic and DFS method.
- A new solution representation technique is designed to accommodate the problem characteristic.
- The OBL and VND techniques are embedded into HTLBO on purpose of enhancing algorithm's performance.

The remainder of this work is organized as follows. Section 2 presents the problem description and the mathematical program model. Section 3 illustrates the theory of TLBO, and Section 4 establishes the proposed HTLBO by combing TLBO with some efficient techniques. Computational experiments are discussed in Section 5. Finally, conclusions and future researches are presented in Section 6.

2. Problem Formulation

2.1. Problem Description

In this paper, a traveler is assumed to visit all tourist attractions in a set of cities. The railway network associated with these cities is known with certainty, and decisions should be made are: (1) the visiting sequence of different cities, (2) the railway route between different cities, and (3) the traveling routes in each city. The objective to be optimized is the total traveling time. Some premises are stated as follows:

- The traveler is assumed to travel by railway between any two cities, while the roads account for the traveling inside a city.
- The time span of the railway network as well as the road network is predefined.
- Once the traveler starts to visit one city, it is assumed that he will reach the city through one of its' railway stations, then go through all tourist attractions by a certain order and leave from one railway station.
- The initial and finale position of the travel is assumed to be at one railway station of then a city.

2.2. Mathematical Model

The relevant notations used for describing the considered problem are collected as follows:

	Parameters
C	Set of cities, $c \in C$
S^c	Set of railway stations in city c
T^c	Set of tourist attractions in city c
V^c	$V^c = S^c \cup T^c$ $V^c = S^c$, set of railway stations and tourist attractions in city c
E^c	$E^c = \{(i,j) \mid i,j \in V^c\}$, set of directed edges in city c
G^c	$G^c = (V^c, E^c)$, the directed network graph of city c
t^c_{ij}	Time span from position i to j in city c, where $i,j \in V^c$ and $(i,j) \in E^c$
V^r	$V^r = \underset{c \in C}{\cup} S^c$, set of railway stations in all cities
E^r	$E^r = \{(i,j) \mid i,j \in V^r\}$, set of directed edges of railway network
G^r	$G^r = (V^r, E^r)$, the directed network graph of railway stations in all cities
t^r_{ij}	Time span from station i to j, where $i,j \in V^r$ and $(i,j) \in E^r$
	Decision variables
x^c_{ij}	$x^c_{ij} \in \{0,1\}$, if the traveler visits tourist attraction j right after i in city c, $x^c_{ij} = 1$, otherwise $x^c_{ij} = 0$
y^r_{ij}	$x^c_{ij} \in \{0,1\}$, if the traveler visits station j right after station i, $x^c_{ij} = 1$, otherwise $x^c_{ij} = 0$

By virtue of above problem description and notation definitions, the considered travel route optimization problem can be modeled as follows:

$$\min Z = \sum_{c\in C}\sum_{i\in V^c}\sum_{j\in V^c} x_{ij}^c \bullet t_{ij}^c + \sum_{i\in V^r}\sum_{j\in V^r} y_{ij}^r \bullet t_{ij}^r \quad (1)$$

$$\sum_{j\in V^c} x_{ij}^c = 1, i \in T^c, c \in C \quad (2)$$

$$\sum_{i\in V^c} x_{ij}^c = 1, j \in T^c, c \in C \quad (3)$$

$$\sum_{t\in S^c}\sum_{j\in V^c} x_{ij}^c = 1, c \in C \quad (4)$$

$$\sum_{i\in V^c}\sum_{j\in S^c} x_{ij}^c = 1, c \in C \quad (5)$$

$$x_{ij}^c \in \{0,1\}, i \in V^c, j \in V^c, c \in C \quad (6)$$

$$y_{ij}^r \in \{0,1\}, i \in V^r, j \in V^r \quad (7)$$

The objective is described in Equation (1), which aims to minimize the total traveling time between and in different cities. Constraints (2) and (3) indicate that each tourist attraction in one city can be visited exactly only one time. Constraints (4) and (5) indicates that travelers enter into and leave from each destination city only through one railway station. Constraints (6) and (7) are the value range constraints of all decision variables.

3. TLBO Algorithm

TLBO algorithm is established by simulating behaviors in a classroom. It uses a group of individual solutions to proceed to a satisfactory solution. The evolution of TLBO includes two parts: 'Teacher Phase' and 'Learner Phase' Figure 2 illustrates a schematic diagram of TLBO.

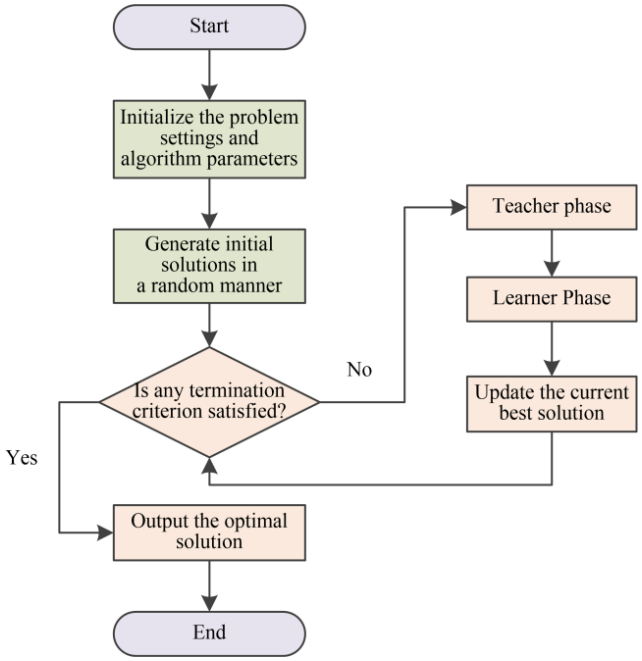

Figure 2. Schematic diagram of TLBO.

Details of some critical steps in TLBO are described as follows:

(1) Initialization

Let N represents the population size and $x_i = (x_{i1}, \cdots, x_{id}, \cdots, x_{iD})$ be the i-the solution vector in the current population ($i = 1, 2, \cdots, N$). Individual x_i is generated by taking the following equation:

$$x_{id} = x_{id}^l + rand(0,1) \bullet (x_{id}^u - x_{id}^l) \tag{8}$$

where x_{id} represents the d-th value of the solution vector x_i and $[x_d^l, x_d^u]$ is range of dimension d. Notation rand$(0,1)$ represents a uniform random number from interval [0,1].

(2) Teacher Phase

The mean position of the current population, noted as $x^m = (m_1, \cdots, m_d, \cdots, m_D)$, is calculated and then to be used for solution update. Besides, the best solution found so far by TLBO is denoted as the teacher individual $x^{teacher}$. Given a parent individual x^{old}, the neighborhood search in this stage is implemented as follows

$$x^{new} = x^{old} + rand(0,1) \bullet (x^{teacher} - \mathrm{TF} \bullet x^m) \tag{9}$$

where the teaching factor $TF \in \{1, 2\}$. A greedy selection strategy is used to update the candidate solutions. In others words, the better solutions between x^{new} and x^{old} will be retained.

(3) Learner Phase

In the so-called "Learner Phase", TLBO update a solution by comparing the position difference between the candidate solution and a randomly selected solution in current population. Specifically, given an individual x^{old}, a random solution $x^{rand}(x^{rand} \neq x^{old})$ is selected. Then, the neighborhood search in this stage is carried out by taking the following equation:

$$x^{new} = \begin{cases} x^{old} + rand(0,1) \bullet (x^{rand} - x^{old}), if\ f(x^r) < f(x^{old}) \\ x^{old} + rand(0,1) \bullet (x^{old} - x^{rand}),\ otherwise \end{cases} \tag{10}$$

where notation $f(\cdot)$ represents the objective function to be optimized. Similarly, the greedy selection strategy is also used to determine the offspring individual.

4. HTLBO Algorithm

4.1. DFS Technique

In the considered problem, the traveler plans to reach different cities by virtue of the railway network. In such a context, HTLBO algorithm uses DFS technique to determine the shortest route between any two stations. DFS is a kind of tree search method in nature. It starts at the root node and explores as far as possible along each branch before backtracking [25].

To better illustrate the search process of DFS, an example is presented here. Figure 3 presents the schematic diagram of a directed railway network, and Figure 4 provides the route search process from station 1 to station 5. The total number of routes is 7, and the shortest route is "Station 1 → Station 2 → Station 3 → Station 5". The path length is 2.20. In addition, Table 1 summarizes the shortest route and path length between any two stations in the current railway network.

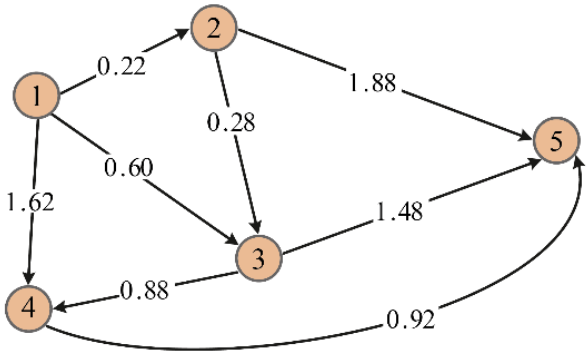

Figure 3. Schematic diagram of a directed railway network.

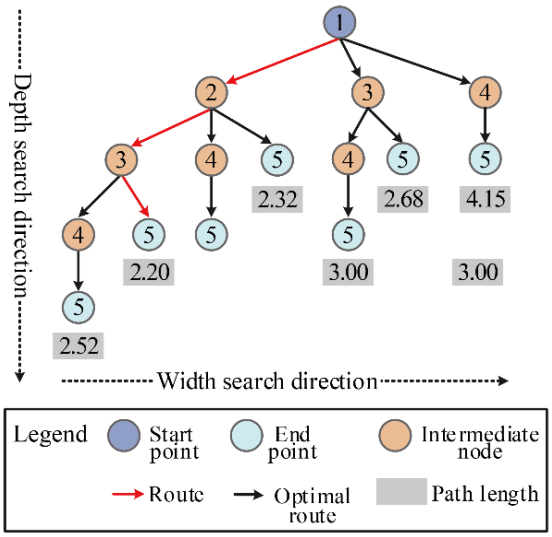

Figure 4. Schematic diagram of the DFS method.

Table 1. Shortest route information in the proposed directed railway network.

Starting Point	Target	Path	Shortest Distance
1	2	1→2	0.22
1	3	1→3	0.60
2	3	2→3	0.28
1	4	1→4	1.62
2	4	2→4	0.92
3	4	3→4	0.88
1	5	1→2→3→5	2.20
2	5	2→5	1.88
3	5	3→5	1.48
4	5	4→5	0.92

4.2. Encoding and Decoding

As TLBO is initially designed for continuous problems, a new solution representation technique is designed to accommodate the problem characteristic and used for the algorithm deployment.

Some details associated with coding process are summarized as follows. Given city se C, tourist attractions set T^c and railway stations set S^c ($c \in C$), HTLBO adopts three layers of real number to represent a solution of the considered problem. The value range of each layer is set to [0,1]. The first layer is used to determine the traveling order of all cities and its encoding length is $|C|$. The second layer is used to determine the station index the traveler entering into and leaving from each city. The corresponding encoding length is

$2 \cdot |C|$. The third layer is used to determine the traveling order of all tourist attractions in each city and the code length is $|\cup_{c \in C} S^c|$.

The decoding process includes two parts: ranked-order value (ROV) and roulette wheel rules [26,27]. ROV rule is used to convert a real number array into an integer sequence. HTLBO adopt ROV rule to determine the traveling order of all cities, as well as the tourist attractions visiting sequence in each city (see Figure 5a). The roulette wheel rule is used to determine the station index when traveler entering into or leaving from a city (see Figure 5b). The implementation process of the above two rules are illustrated in Figure 5.

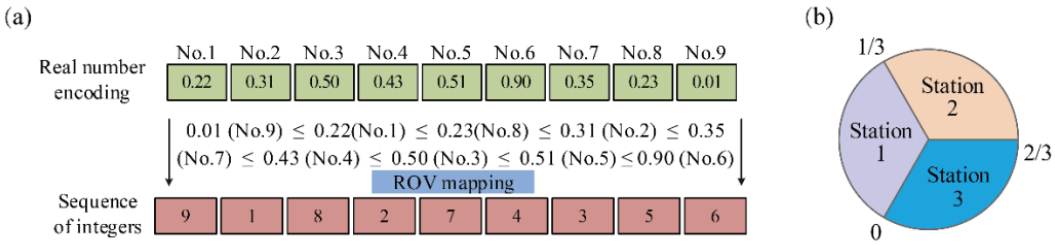

Figure 5. Schematic diagram of the ROV and roulette wheel rules: (**a**) traveling order and tourist attractions visiting sequence in each city; (**b**) station index when traveler entering into or leaving from a city

By virtue of above preparations, the decoding process of a solution array for the considered problem is summarized as follows: First, ROV rule is used to obtain the traveling order of all cities. Then, roulette wheel rule is utilized to determine the station index when entering into or leaving from a city. Finally, the ROV rule is used to obtain the visiting order of all the tourist attractions in each city.

To clearly explain the above process, a concrete example is present here. Table 2 provides the set of tourist attractions as well as the set of railway stations in each city. On this basis, the coding length of each layer is 3, 6, and 10, respectively. Given three levels of coding ((0.22, 0.50, 0.31), (0.81, 0.30, 0.12, 0.51, 0.71, 0.31), (0.22, 0.31, 0.50, 0.43, 0.51, 0.90, 0.35, 0.23, 0.01, 0.99)), Figure 6 presents the corresponding decoding process.

Table 2. Information of a concrete instance.

City Index	Set of Tourist Attractions	Set of Railway Stations
1	1,2,3	1,2
2	4,5,6,7	3,4,5
3	8,9,10	6,7

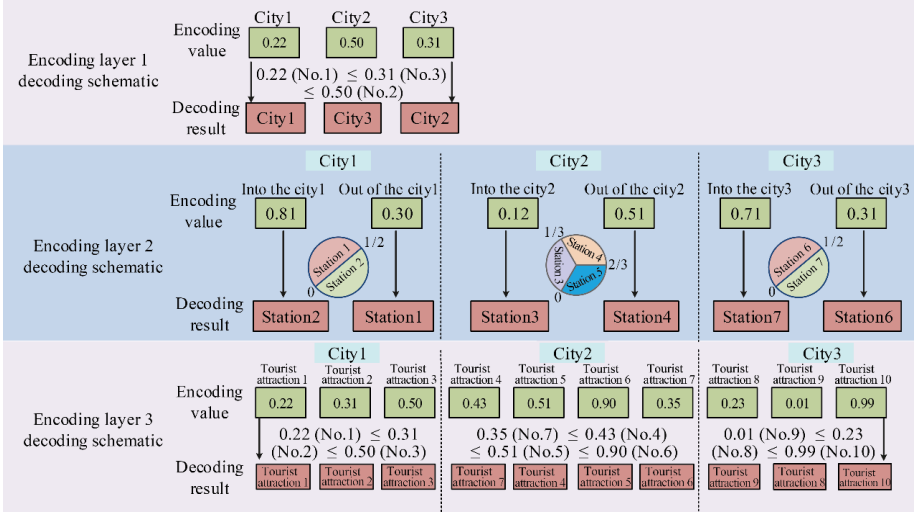

Figure 6. The classification flow chart of creative industry.

The interpretation of such a solution array is described as follow:

- First, the traveler will enter in city 1 via station 2, then visit tourist attractions 1→2→3, and finally leave from the current city from station 1.
- Secondly, the traveler will enter in city 3 via station 7, then visit tourist attractions 9→8→10, and finally leave from the current city from station 6.
- Finally, enter city 2 via station 3, the traversal sequence of tourist attractions is 7→4→5→6, and leave the current city from station 4.

4.3. OBL

OBL is a simple but efficient method to enhance algorithm's performance in swarm computation [28]. In the design of HTLBO, OBL is often used to improve the quality of population in two different ways.

First, OBL is used to generate initial solutions with high qualities. Given population size N and the i-the solution $x_i = (x_{i1}, \cdots, x_{id}, \cdots, x_{iD})$ $(i = 1, 2, \cdots, N)$ in the current population, the reverse solution array $x_i^o = (x_{i1}^o, \cdots, x_{id}^o, \cdots, x_{id}^o)$ of x_i is calculated by:

$$x_{id}^o = x_d^l + x_d^u - x_{id} \tag{11}$$

where $d = 1, 2, \cdots, D$, D represents the problem dimension and interval $[x_d^l, x_d^u]$ is the value range of the d-the dimension decision variable. The initialization of HTLBO is implemented by taking the following steps:

Step 1. Let $i \leftarrow 1$ and $d \leftarrow 1$, then go to Step 2.
Step 2. If $i \leq N$, then go to Step 3; otherwise, go to Step 7.
Step 3. If $d \leq D$ is satisfied, go to Step 4; otherwise, go to Step 5.
Step 4. Let $x_{id} = x_{id}^l + \text{rand}(0,1) \cdot (x_{id}^u - x_{id}^l)$ and $x_{id}^o = x_d^l + x_d^u - x_{id}$, respectively; then, go to Step 5.
Step 5. Let $d \leftarrow d + 1$, if $d > D$ holds, let $d \leftarrow 1$ and go to Step 6; otherwise, go to Step 3.
Step 6. Let $i \leftarrow i + 1$, and then go to Step 2.
Step 7. Select N best solutions among $\{x_1, \cdots, x_N, x_i^o, \cdots, x_N^o\}$ to form the initial population.

Second, OBL is employed to explore the search space when the new generated solution is not superior to the candidate one in "Teacher Phase" or "Learner Phase". For the candidate solution, the reverse solution is defined as follows:

$$x_{id}^o = \begin{cases} x_d^l + x_d^u - x_{id}, if\ rand(0,1) < 0.5 \\ rand(0,1) \bullet (\widetilde{x}_d^l - \widetilde{x}_d^u) - x_{id},\ otherwise \end{cases} \tag{12}$$

where $[\widetilde{x}_d^l, \widetilde{x}_d^u]$ is the dynamic boundary of the current population, i.e., $\widetilde{x}_d^l = \min\{x_{id} \big| i = 1, 2, \cdots, N\}$ and $\widetilde{x}_d^u = \max\{x_{id} \big| i = 1, 2, \cdots, N\}$. Notation $\text{rand}(0,1)$ represents a random number in interval [0,1].

4.4. Local Search

To better adopt TLBO for the considered problem, a local search method based on VND is designed in this subsection [29]. Given an initial solution, VND is performed in a deterministic way and it explores the search space by virtue of different mutation operators. In HTLBO, Gaussian mutation (GM), swap, reversion and insertion operators utilize to generate mutant solution array of a candidate one [30,31]. Details of each mutation operator are summarized as follows:

- GM: given the d-the dimension of solution array (denoted as x_{id}), the update process is implemented according to probability P_{GM}. The mathematical equations are:

$$x_{id} = \begin{cases} GM(x_{id}, \sigma_d)\ if\ rand(0,1) < P_{GM} \\ x_{id} \quad otherwise \end{cases} \tag{13}$$

$$\sigma_d = \frac{x_{id}^u - x_{id}^l}{\sigma_{scale}} \quad (14)$$

where $GM(x_{id}, \sigma)$ denotes Gaussian sampling with mean x_{id} and variance σ_d. Notation σ_d is calculated by range $[x_{id}^l, x_{id}^u]$ and the scaling factor σ_{scale} is a scaling factor. According to some related studies, P_{GM} is set to the reciprocal of the total length of the code and σ_{scale} is set to 20.

- Swap: randomly select two positions in a candidate solution array, and exchange values on these two positions (see Figure 7a).
- Reversion: randomly select two positions in a candidate solution array, and reverse values between these two positions (see Figure 7b).
- Insertion: randomly select two positions in a candidate solution array, and insert the value on a position before the other position (see Figure 7c).



Figure 7. Schematic diagram of the neighborhoods (**a**) Swap (**b**) Reversion (**c**) Insertion.

Gaussian mutation can be utilized to update each lay of the candidate solution array, while the other three operators are applied to update the first, as well as the third layer. Given a candidate solution array x_i, a random permutation (denoted as $< \Omega_1, \Omega_2, \Omega_3, \Omega_4 >$) of above-mentioned four operators are first generated. With a predefined iteration number $LS_{\max}$, the local search based on VND is carried out by taking the following steps:

Step 1. Let $i \leftarrow 1$ and $t \leftarrow 1$, then go to Step 2.
Step 2. Generate a variant solution x_i' by virtue of operator Ω_i, and then go to Step 3.
Step 3. If x_i' is better than x_i, set $x_i \leftarrow x_i'$, $i \leftarrow 1$ and $t \leftarrow 1$, respectively, and then go to Step 2; otherwise, set $t \leftarrow t + 1$ and go to Step 4.
Step 4. If $t < LS_{\max}$, go to Step 2; otherwise, set $t \leftarrow 1$ and $i \leftarrow i + 1$, respectively, and then go to Step 5.
Step 5. If $i < 4$, go to Step 2; otherwise, terminate the iteration and output x_i'.

4.5. Flow of HTLBO

Based on previous descriptions, Figure 8 illustrates the overall procedure of HTLBO metaheuristic. As it can be noticed, HTLBO contains three significant components, namely initialization, "Teacher Phase" and "Learner Phase". Detailed descriptions are stated as follows:

- In initialization, DFS is applied to determine optimal route of any two stations in the railway network. Besides, HTLBO adopts OBL technique to generate initial population with high qualities.
- In "Teacher Phase", the original neighbor search of TLBO is retained, and OBL technique is utilized to improve solution performance.
- In "Learner Phase", the original neighbor search is retained, and the VND-based local search coupled with four mutation operators is utilized to enhance the algorithm's exploitation ability.

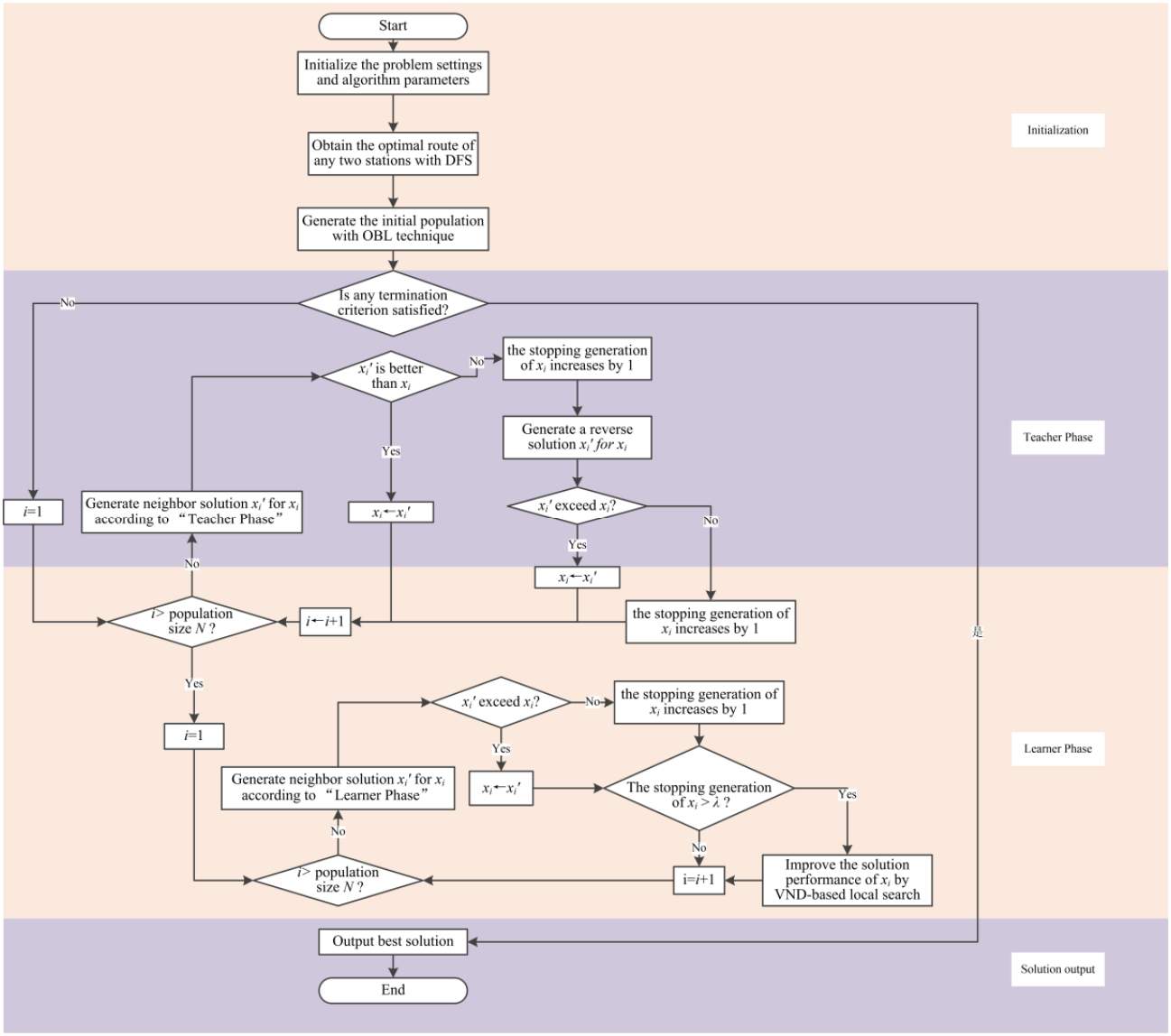

Figure 8. Flow chart of HTLBO.

5. Case Study

5.1. Experiment Preparation

All optimization algorithms were implemented in MATLAB 2016a platform and run on a computer with 1.6 GHz, 8 GB memory intel(R) core (TM)i7-10510U CPU (Central Processing Unit).

The optimization of urban tourist routes along the Zhejiang section of the Shanghai–Hangzhou–Ningbo Railway is taken as an object, and the railway network is used as a

convenient link between cities. The Shanghai–Hangzhou–Ningbo Railway is an important railway line built in the late Qing Dynasty that runs through the important economic towns in the south of the Yangtze River. It has a history of more than 120 years. The use of the Shanghai–Hangzhou–Ningbo Railway to develop tourism has a history of nearly a hundred years. Using ancient architectural tourism to drive the development of railways can not only promote the development of tourism in surrounding cities, but also has historical significance and time value for railway protection.

The Zhejiang section of the Shanghai–Hangzhou–Ningbo Railway encompasses 4 target cities, Jiaxing, Hangzhou, Shaoxing, as well as Ningbo, with 10 existing stations and 52 national key cultural relics under construction in cities along the line. The names of railway stations are replaced by capital letters in Table 3, and the names of tourist attractions are replaced by numbers in Table 4. For example, In Table 3 the 'A' refers to Jiashan station which is shown in the table key below. Table 5 summarizes the location information of 10 stations and 52 tourist attractions. Table 6 shows the railway travel direction and traffic time of adjacent stations between cities. "–" means that there is no train passing in the current direction. In addition, the travel time between any two locations in each city can be obtained through the Gaode map.

Table 3. The code name of railway station.

Code Name	A	B	C	D	E	F	G	H	I	J
Station Name	Jiashan	Jiaxing	Haining	Hangzhou	Shaoxing	Shaoxing East	Shangyu	Yuyao	Zhuangqiao	Ningbo

Table 4. The code name of tourist attractions.

Code	City/Tourist Attractions	Code	City/Tourist Attractions
	Jiaxing		**Hangzhou**
1	The site of the First Congress of the Communist Party of China, Nanhu, Jiaxing	9	Zhijiang University Site
		10	Jianqiao Central Aviation School Site
2	Wang Guowei's Former Residence	11	Former Residence of Zhang Taiyan
3	Former Residence of Mao Dun	12	Ma Yinchu Former Residence
4	Jiaxing Wensheng Monastery and Catholic Church	13	The former site of Zhejiang Industrial Bank
5	Moshi Manor	14	Xiling Seal Society
6	Chen Ge Old House	15	Wenlan Court
7	Qiyuan	16	Hu Qingyutang (Dajing Lane)
8	Yanguan Seawall and Temple of the sea	17	Hero Tower
	Ningbo	18	Zhakou White Tower
36	The former site of the East Zhejiang Anti-Japanese Base, Yuyao	19	Lingyin Temple Two Stone Pagodas and Two Sutras
		20	Baochu Tower
37	The former site of the East Zhejiang Anti-Japanese Base, Cixi	21	Phoenix Temple
38	Chiang's Former Residence	22	Liuhe Tower
39	Longshan Yu's old house complex Cao'e Temple	23	Stone Pagoda of Puqing Temple
40	Jiangbei Catholic Church		**Shaoxing**
41	Qianye Club	24	Former Residence of Lu Xun
42	Jintang School Site	25	Former Residence of Qiu Jin, Zhejiang
43	Lin House	26	Cai Yuanpei's Former Residence
44	Former residence of Wang Shouren	27	Cao'e Temple
45	Tianyi Pavilion	28	Ma Yinchu Former Residence
46	Qing'an Hall	29	Xu Xilin's Former Residence
47	Cicheng Ancient Buildings	30	Datong School
48	Baoguo Temple	31	Chunhui Middle School Site
49	Ayuwang Temple	32	Lv House
50	Tiantong Temple	33	Qingteng bookstore
51	Tianning Temple	34	Lanting
52	Erling Tower	35	Shunwang Temple

Table 5. Location information of four cities.

City	Code	Longitude	Latitude	Code	Longitude	Latitude	Code	Longitude	Latitude
Jiaxing	A	120.91	30.85	2	120.53	30.40	6	120.54	30.41
	B	120.76	30.77	3	120.49	30.74	7	120.94	30.52
	C	120.68	30.54	4	120.75	30.78	8	120.54	30.40
	1	120.76	30.76	5	121.02	30.70			
Hangzhou	D	120.18	30.25	14	120.13	30.25	20	120.14	30.26
	9	120.12	30.19	15	120.14	30.25	21	120.17	30.25
	10	120.22	30.32	16	120.16	30.24	22	120.13	30.20
	11	120.00	30.29	17	119.73	30.23	23	119.76	30.36
	12	120.16	30.26	18	120.14	30.20			
	13	120.17	30.25	19	120.10	30.24			
Shaoxing	E	120.58	30.02	26	120.58	30.01	31	120.92	30.03
	F	120.85	30.06	27	120.87	29.99	32	120.57	30.01
	G	120.85	30.00	28	120.85	29.63	33	120.57	30.00
	24	120.58	30.00	29	120.53	30.06	34	120.49	29.93
	25	120.58	29.99	30	120.57	30.01	35	120.70	29.79
Ningbo	H	121.16	30.06	40	121.56	29.88	47	121.45	29.99
	I	121.53	29.93	41	121.55	29.87	48	121.52	29.98
	J	121.53	29.86	42	121.42	30.19	49	121.74	29.85
	36	121.08	29.89	43	121.54	29.87	50	121.79	29.81
	37	121.57	30.12	44	121.15	30.05	51	121.54	29.88
	38	121.28	29.68	45	121.54	29.87	52	121.68	29.77
	39	121.55	30.09	46	121.56	29.87			

Table 6. Location information of four cities.

Station	A	B	C	D	E	F	G	H	I	J
A	–	0.22	0.60	1.62	–	–	–	–	–	–
B	–	–	0.28	0.92	1.88	–	2.32	2.95	–	3.77
C	–	–	–	0.88	1.48	–	1.92	2.53	–	3.37
D	–	–	–	–	0.92	–	1.33	1.75	–	2.67
E	–	–	–	–	–	–	0.35	0.73	–	1.48
F	–	–	–	–	–	–	–	–	0.45	0.43
G	–	–	–	–	–	–	–	0.48	–	1.47
H	–	–	–	–	–	–	–	–	–	0.70
I	–	–	–	–	–	–	–	–	–	0.17
J	–	–	–	–	–	–	–	–	–	–

5.2. Parameters Calibration

The calibration of algorithm parameters in metaheuristic approaches has a significant effect on the solution qualities. In the current research, parameters in HTLBO are tuned by Taguchi method for obtaining solutions with satisfactory performance.

Parameters in HTLBO that have to be tuned include: population size N, local search control threshold λ, local search iterations $LS_{\max}$ and maximum iterations $MaxIt$. On this basis, Table 7 collects all values of parameters at different levels, and the corresponding orthogonal array is displayed in Table 8. Then, the proposed instance is selected for the test. For every trial, HTLBO is run for 30 independent times and the mean result (i.e., $\overline{Z}$) is collected. Based on simulation results in Table 8, the range analysis of HTLBO's parameters is presented in Table 9. Besides, the analysis of variance (ANOVA) is used to examine the statistical significance test of a HTLBO's parameters on the algorithm performance, where the significance level is fixed at 0.05 [32]. Some deductions can be captured as follows:

- As can be noticed in Table 9, the rank of parameters impacting HTLBO's performance is $LS_{\max}$, N, λ, and $MaxIt$.

- Parameters of N λ and $LS_{\max}$ have a significant impact on HTLBO's performance since their p values are less than 5% (see Table 10).
- The levels with highest values of $\overline{Z}$ in Table 9 are to be selected as the optimal levels: $N = 50$, $\lambda = 100$, $LS_{\max} = 20$, and $MaxIt = 2000$.

Table 7. Parameters and levels.

Level	Factor			
	N	λ	$LS_{\max}$	$MaxIt$
1	30	50	10	1000
2	40	100	20	1500
3	50	150	30	2000

Table 8. Experimental design and simulation results.

Trial	Factor				$\overline{Z}$	Trial	Factor				$\overline{Z}$
	N	λ	$LS_{\max}$	$MaxIt$			N	λ	$LS_{\max}$	$MaxIt$	
1	1	1	1	1	23.75	15	2	2	3	1	21.90
2	1	1	1	1	22.99	16	2	3	1	2	24.00
3	1	1	1	1	24.06	17	2	3	1	2	23.73
4	1	2	2	2	20.61	18	2	3	1	2	24.20
5	1	2	2	2	21.71	19	3	1	3	2	21.03
6	1	2	2	2	22.24	20	3	1	3	2	21.80
7	1	3	3	3	23.62	21	3	1	3	2	19.90
8	1	3	3	3	24.15	22	3	2	1	3	23.18
9	1	3	3	3	21.95	23	3	2	1	3	22.11
10	2	1	2	3	21.59	24	3	2	1	3	20.69
11	2	1	2	3	20.71	25	3	3	2	1	19.74
12	2	1	2	3	20.73	26	3	3	2	1	21.90
13	2	2	3	1	20.39	27	3	3	2	1	22.21
14	2	2	3	1	21.87						

Table 9. Range analysis of HTLBO's parameters on mean $\overline{Z}$.

Level	Parameters			
	N	λ	$LS_{\max}$	$MaxIt$
1	68.36	65.52	69.58	66.27
2	66.38	64.90	63.81	66.41
3	64.19	68.50	65.54	66.24
Delta	4.17	3.60	5.76	0.16
Rank	2	3	1	4
Optimal level	3	2	2	3

Table 10. ANOVA table for parameters calibration.

Parameters	Sum of Squares	Degree of Freedom	Mean of Squares	F Distribution	p Value
N	8.70	2	4.35	5.12	0.0174
λ	7.41	2	3.70	4.36	0.0286
$LS_{\max}$	17.50	2	8.75	10.29	0.0010
$MaxIt$	0.02	2	0.01	0.01	0.9909
Error	15.30	18	0.85		

5.3. Analysis of Test Results

With respect to the real-world case, the proposed HTLBO is evaluated by being compared with three metaheuristic approaches: basic TLBO, GA, and PSO [33–35]. The reasons for such a choice of these benchmark algorithms are stated as follows. First, the comparison between HTLBO and TLBO is beneficial to reveal the performance of OBL- and VND-based modifications. Second, TLBO is treated as one of most powerful metaheuristic algorithms by a great many researched in different fields. Third, PSO and GA have been successfully introduced to TSP and its variants, which are similar to the considered problem in current research. Therefore, such a comparison is meaningful to some extent. The population size and maximum iterations of benchmark algorithms are set to the same values as HTLBO. Besides, the other parameters are referred to previous research work.

Based on above parameter settings, the experiments are carried out and simulation results are collected in Table 11. Each algorithm is run for 30 independent times. Some metrics are calculated, where Z^{opt}, Z^{wor}, Z^{mean} and Z^{std}, respectively, represent optimal value, worst value, mean value and standard deviation. Besides, the average percentage relative deviation $APRD$ is computed by taking the flowing equation:

$$APRD = \frac{Z^{mean} - Z^{lb}}{Z^{lb}} \times 100\% \tag{15}$$

where Z^{mean} is the mean value of a selected algorithm in 15 independent runs and Z^{lb} demotes the optimal value obtained by all tested algorithms. Besides, Figure 9 compares the optimal evolution curves of the four algorithms, and the optimal travel routes found by each algorithm are presented in Table 12.

Table 11. Comparison of simulation results.

Algorithm	Z^{opt}	Z^{mean}	Z^{wor}	Z^{std}	$APRD$/%	CPU/Seconds
HTLBO	19.74	20.45	22.58	0.71	3.59	477.26
TLBO	20.79	22.17	23.11	1.83	12.31	401.63
GA	20.85	22.04,	23.86	1.24	11.65	438.40
PSO	20.57	21.45	23.61	2.13	8.66	394.11

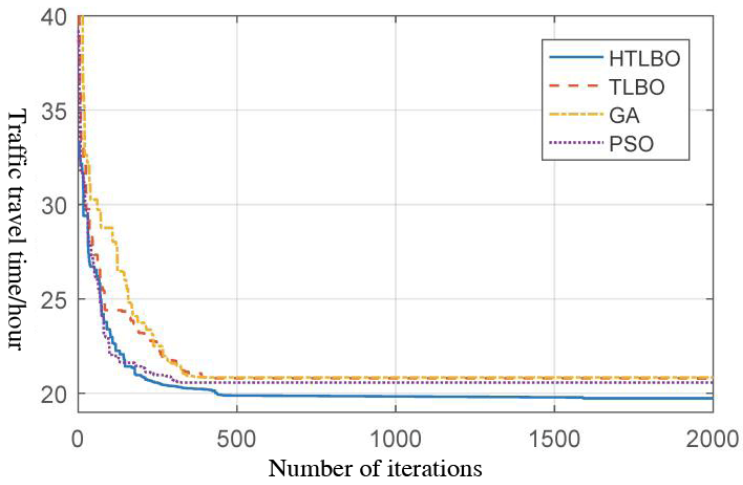

Figure 9. Comparison of optimal evolutionary curves.

Table 12. Optimal routes obtained by fours algorithms.

Algorithm	Tourist Route
HTLBO	**B**→4→3→2→8→6→7→5→1→**B**→**D**→16→21→13→28→20→15→14→18→22→9→19→17→23→11→10→**D**→**E**→26→24→25→33→30→32→29→34→35→28→31→27→**G**→**H**→44→42→37→39→36→48→47→38→52→50→49→46→40→41→51→45→43→**J**
TLBO	**C**→6→8→2→3→4→1→5→7→C→**D**→16→18→22→9→14→15→20→19→17→23→11→10→28→13→21→**D**→E→32→29→33→24→30→26→25→34→35→28→27→31→**F**→**J**→43→41→51→45→47→48→37→39→37→42→44→38→52→50→49→46→40→**I**
GA	**C**→8→2→6→3→7→5→1→4→**B**→**D**→18→22→9→15→16→21→13→28→20→14→19→17→23→11→10→**D**→**E**→32→26→24→25→33→30→29→34→35→28→27→31→**F**→**J**→45→40→49→50→52→38→43→46→41→51→48→37→39→36→42→47→44→**H**
PSO	**B**→3→2→8→6→7→5→1→4→**B**→**D**→21→16→18→22→9→17→23→11→19→20→13→14→15→28→10→**D**→**E**→29→32→26→33→25→24→30→34→35→28→31→27→**G**→**H**→44→38→47→42→37→39→36→48→40→46→41→51→45→43→52→50→49→**I**

As could be seen, HTLBO yields the best solution performance of the four algorithms in terms of objective metrics (i.e., Z^{opt}, Z^{wor} and Z^{mean}) and $APRD$. The $APRD$ value of HTLBO is 3.59%, which is significantly superior the other three benchmark algorithms. PSO performs second to HTLBO with $APRD$ of 8.66 %, followed by GA whose $APRD$ amounts to 11.65%. The original TLBO shows weakest performance with $APRD$ of 12.31%. In other words, the applications of OBL- and VND-based modifications have a significant improvement on the algorithm performance. With respect to Z^{std} metric, HTLBO yields the best performance of these four algorithms whose value is 0.7. Thus, the proposed hybrid metaheuristic performs more robustly than the other three algorithms. In addition, the mean computation time (denoted as CPU) of all tested algorithms are very close to each other.

To examine the statistical significance of simulation results, a least significant difference (LSD) test is used to analyze the objectives values of four algorithms in all 30 runs [36]. Sum of squares and mean of squares are used to describe the influence from the algorithm section and noise factors. The statistical results are collected in Table 13, where the significance level value is fixed at 0.05. As can be seen, the p-value in current table is pretty close to zero. Thus, it could be concluded that the proposed HTLBO exceed the other benchmark algorithms in a statistically significant manner. Furthermore, the LSD interval at a 95% level is depicted in Figure 10. As can be noticed, the proposed HTLBO outperforms other three metaheuristics in a statistical manner.

Table 13. One-way ANOVA table.

Source	Sum of Squares	Degree of Freedom	Mean of Squares	F Distribution	p-Value
Algorithm	56.104	3	18.7014	23.19	7.87×10^{-12}
Error	93.554	116	0.8065		
Total	149.659	119			

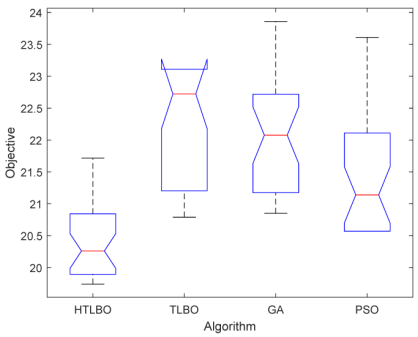

Figure 10. LSD test of for all algorithms.

6. Conclusions

In this paper, a real-world travel route optimization problem, alongside the urban railway line, is investigated from the perspectives of practical application. First, a mathematical model is formulated with an objective of minimizing the total traveling time. Then, a hybrid metaheuristic named HTLBO is proposed for solution generation. In HTLBO, DFS is employed to obtain the optimal routes of any two stations in railway network, and a three-level coding method is designed to represent a solution to the considered problem. Besides, OBL- and VND-based techniques are utilized to enhance algorithm's performance. Finally, a case study is presented, and the proposed HTLBO is compared with TLBO, GA, and PSO. The effectiveness of HTLBO is validated by simulations on the practical case in terms of convergence and robustness.

One interesting topic of future research work is to extend HTLBO to more complicated travel route optimization problems and some other similar practical applications. Another research direction is to establish a new mathematical model of the current problem with the consideration of some other factors, such as cost and low-carbon.

Author Contributions: Data curation, C.L. and C.H.; Formal analysis, C.H.; Methodology, C.L. and Q.Z.; Project administration, F.L.; Supervision, F.L.; Writing—original draft, C.L.; Writing—review & editing, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China: [No. 52078107], National Science Foundation of Zhejiang Province: [LY18E080033], the Fundamental Research Funds for the Central Universities of China: [No. N2011002].

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

References

1. Pratt, S. Potential Economic Contribution of Regional Tourism Development in China: A Comparative Analysis. *Int. J. Tour. Res.* **2015**, *17*, 303–312. [CrossRef]
2. Zuo, B.; Huang, S. Revisiting the Tourism-Led Economic Growth Hypothesis: The Case of China. *J. Travel Res.* **2017**, *57*, 151–163. [CrossRef]
3. Chen, L.; Thapa, B.; Yan, W. The Relationship between Tourism, Carbon Dioxide Emissions, and Economic Growth in the Yangtze River Delta, China. *Sustainability* **2018**, *10*, 2118. [CrossRef]
4. Xie, X.F.; Liu, J. Multiagent Optimization System for Solving the Traveling Salesman Problem (TSP). *IEEE Trans. Syst. Man Cybern. Part B* **2009**, *39*, 489–502. [CrossRef]
5. Lendl, S.; Ćustić, A.; Punnen, A.P. Combinatorial Optimization with Interaction Costs: Complexity and Solvable Cases. *Discrete Optim.* **2019**, *33*, 101–117. [CrossRef]
6. Li, J.; Xu, L.; Tang, L.; Wang, S.; Li, L. Big Data in Tourism Research: A Literature Review. *Tour. Manag.* **2018**, *68*, 301–323. [CrossRef]
7. Tilk, C.; Irnich, S. Dynamic Programming for the Minimum Tour Duration Problem. *Transp. Sci.* **2017**, *51*, 549–565. [CrossRef]
8. Poikonen, S.; Golden, B.; Wasil, E.A. A Branch-and-Bound Approach to the Traveling Salesman Problem with a Drone. *INFORMS J. Comput.* **2019**, *31*, 335–346. [CrossRef]
9. Goerigk, M.; Grün, B.; Heßler, P. Branch and Bound Algorithms for the Bus Evacuation Problem. *Comput. Oper. Res.* **2013**, *40*, 3010–3020. [CrossRef]
10. Liao, Z.; Zheng, W. Using a Heuristic Algorithm to Design a Personalized Day Tour Route in a Time-Dependent Stochastic Environment. *Tour. Manag.* **2018**, *68*, 284–300. [CrossRef]
11. Zheng, W.; Liao, Z.; Qin, J. Using a Four-Step Heuristic Algorithm to Design Personalized Day Tour Route within a Tourist Attraction. *Tour. Manag.* **2017**, *62*, 335–349. [CrossRef]
12. Pop, P.C.; Zelina, I.; Lupşe, V.; Sitar, C.P.; Chira, C. Heuristic Algorithms for Solving the Generalized Vehicle Routing Problem. *Int. J. Comput. Commun. Control* **2011**, *6*, 158–165. [CrossRef]
13. Lu, Y.; Benlic, U.; Wu, Q. A Hybrid Dynamic Programming and Memetic Algorithm to the Traveling Salesman Problem with Hotel Selection. *Comput. Oper. Res.* **2018**, *90*, 193–207. [CrossRef]
14. Ritzinger, U.; Puchinger, J.; Hartl, R.F. Dynamic Programming Based Metaheuristics for the Dial-a-Ride Problem. *Ann. Oper. Res.* **2016**, *236*, 341–358. [CrossRef]
15. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evolut. Comput.* **1997**, *1*, 67–82. [CrossRef]
16. Koehler, G.J. Conditions That Obviate the No-Free-Lunch Theorems for Optimization. *INFORMS J. Comput.* **2007**, *19*, 273–279. [CrossRef]

17. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems. *Comput. Aided Design* **2011**, *43*, 303–315. [CrossRef]
18. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large Scale Problems. *Inf. Sci.* **2012**, *183*, 1–15. [CrossRef]
19. Rao, R.V.; Patel, V. An Elitist Teaching-Learning-Based Optimization Algorithm for Solving Complex Constrained Optimization Problems. *Int. J. Ind. Eng. Comput.* **2012**, *3*, 535–560. [CrossRef]
20. Črepinšek, M.; Liu, S.H.; Mernik, L. A Note on Teaching–Learning-Based Optimization Algorithm. *Inf. Sci.* **2012**, *212*, 79–93. [CrossRef]
21. Chen, W.; Panahi, M.; Khosravi, K.; Pourghasemi, H.R.; Rezaie, F.; Parvinnezhad, D. Spatial Prediction of Groundwater Potentiality Using Anfis Ensembled with Teaching-Learning-Based and Biogeography-Based Optimization. *J. Hydrol.* **2019**, *572*, 435–448. [CrossRef]
22. Li, S.; Gong, W.; Yan, X.; Hu, C.; Bai, D.; Wang, L.; Gao, L. Parameter Extraction of Photovoltaic Models Using an Improved Teaching-Learning-Based Optimization. *Energy Convers. Manag.* **2019**, *186*, 293–305. [CrossRef]
23. Rao, R.V. *Teaching-Learning-Based Optimization (TLBO) Algorithm and Its Engineering Appli-Cations*; Springer International Publishing: Cham, Switzerland, 2016.
24. Gogna, A.; Tayal, A. Metaheuristics: Review and Application. *J. Exp. Theor. Artif. Intell.* **2013**, *25*, 503–526. [CrossRef]
25. Bazin, A. A Depth-First Search Algorithm for Computing Pseudo-Closed Sets. *Discrete Appl. Math.* **2018**, *249*, 28–35. [CrossRef]
26. Zhang, Y.; Zhou, J.; Sun, L.; Mao, J.; Sun, J. A Novel Firefly Algorithm for Scheduling Bag-of-Tasks Applications under Budget Constraints on Hybrid Clouds. *IEEE Access* **2019**, *7*, 151888–151901. [CrossRef]
27. Chung, T.P.; Fu, Q.; Liao, C.J.; Liu, Y.T. Multiple-Variable Neighbourhood Search for the Single-Machine Total Weighted Tardiness Problem. *Eng. Optim.* **2017**, *49*, 1133–1147. [CrossRef]
28. Mahdavi, S.; Rahnamayan, S.; Deb, K. Opposition Based Learning: A Literature Review. *Swarm Evolut. Comput.* **2018**, *39*, 1–23. [CrossRef]
29. de Freitas, J.C.; Penna, P.H.V. A Randomized Variable Neighborhood Descent Heuristic to Solve the Flying Sidekick Traveling Salesman Problem. *Electron. Notes Discret. Math.* **2018**, *66*, 95–102. [CrossRef]
30. Song, S.; Wang, P.; Heidari, A.A.; Wang, M.; Zhao, X.; Chen, H.; He, W.; Xu, S. Dimension Decided Harris Hawks Optimization with Gaussian Mutation: Balance Analysis and Diversity Patterns. *Knowl. Based Syst.* **2021**, *215*, 106425. [CrossRef]
31. Salehi, M.; Maleki, H.R.; Niroomand, S. Solving a New Cost-Oriented Assembly Line Balancing Problem by Classical and Hybrid Meta-Heuristic Algorithms. *Neural Comput. Appl.* **2020**, *32*, 8217–8243. [CrossRef]
32. Chen, W.C.; Nguyen, M.H.; Chiu, W.H.; Chen, T.N.; Tai, P.H. Optimization of the Plastic Injection Molding Process Using the Taguchi Method, Rsm, and Hybrid Ga-PSO. *Int. J. Adv. Manuf. Technol.* **2016**, *83*, 1873–1886. [CrossRef]
33. Haroun, S.A.; Jamal, B.; El Hassani, H. A Performance Comparison of Ga and Aco Applied to TSP. *Int. J. Comput. Appl.* **2015**, *117*, 28–35. [CrossRef]
34. Elloumi, W.; El Abed, H.; Abraham, A.; Alimi, A.M. A Comparative Study of the Improvement of Performance Using a PSO Modified by Aco Applied to TSP. *Appl. Soft Comput.* **2014**, *25*, 234–241. [CrossRef]
35. Yu, Y.; Li, Y.; Li, J. Parameter Identification of a Novel Strain Stiffening Model for Magnetorheological Elastomer Base Isolator Utilizing Enhanced Particle Swarm Optimization. *J. Intell. Mater. Syst. Struct.* **2015**, *26*, 2446–2462. [CrossRef]
36. Derrac, J.; García, S.; Molina, D.; Herrera, F. A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms. *Swarm Evolut. Comput.* **2011**, *1*, 3–18. [CrossRef]