Hindawi
Journal of Advanced Transportation
Volume 2024, Article ID 5964428, 17 pages
https://doi.org/10.1155/2024/5964428

Research Article

Driving Strategy Using an Improved Ant Colony System for Energy-Efficient Train

Chengda Yang, Kun Miao, and Jieyuan Wang

School of Civil Engineering, Central South University, Changsha 410075, Hunan, China

Correspondence should be addressed to Kun Miao; miao98004@csu.edu.cn

Received 24 May 2023; Revised 16 November 2023; Accepted 26 December 2023; Published 18 January 2024

Academic Editor: Zhihong Yao

Optimal energy-efficient train operation optimization is one of the widely studied areas in transportation science, which can significantly reduce energy consumption that accounts for a large proportion of operating costs. In order to adapt to the complex and changeable railway line conditions such as gradient, slope length, and speed limit and avoid the error in tracking speed curve, an optimal driving strategy decision-making (ODSD) model is proposed in this paper. The model considers the non-fixed sequence of driving regimes, and the regimes are directly selected in the discrete micro-subsegments of an equal time-division pattern. To solve this model efficiently, an improved ant colony system algorithm with the difference edges (ACSd) is proposed, which takes the heuristic effect of the difference between the best solutions of two adjacent iterations, i.e., "the difference edges," into account. Additionally, energy-efficient heuristic factor and speed heuristic factor are presented to balance energy saving and speed. The results demonstrate that ACSd performs better than the basic ant colony system algorithm in solving traveling salesman problem (TSP) and provides more flexible driving strategies for the ODSD model.

1. Introduction

Transportation energy consumption of China reached 9.0% of the total energy consumption in 2019 [1], and the railway sector consumed 14 billion kilowatt hours of electricity in 2021 [2]. The train operation consumes so much energy, and thus how to improve energy efficiency in train operation has attracted more attention in recent years. Many energy-saving strategies have been considered for energy efficiency, such as equipment innovation in lightweight vehicle body. Despite the fact that such a hardware modification can promote energy conservation, the improvement on train operations is a highly promising choice for energy saving [3] since it does not require any costs. The improvement may be achieved by solving the train driving regime selection problem for finding an optimal sequence of the driving regimes. An energy-efficient driving model should be developed before solving this problem.

Ichikawa [4] introduced Pontryagin's maximum principle into an energy-efficient driving model and first proposed the analytic method for optimizing driving regimes. Since then, this method has widely been used in the energy-efficient models [5–9]. The analytic method refers to getting the optimal driving regime sequence through strict analytic function derivation upon the optimal control theory. Specifically, the optimal driving regimes of train operation and the sequence of the regimes are derived through the analytic equation constructed by Pontryagin's maximum principle, and then the location of the regime switching point is derived under different constraint conditions.

The optimal driving regime sequence derived is different from the different established energy-efficient models in the literature. For gentle slopes and short intervals, Milroy [10] established an optimization model and derived the optimal driving regime consisting of maximum traction, coasting, and maximum braking. On this basis, Lee et al. [11] thought that for longer operating ranges, the optimal driving regime should include cruise regime. As for the combination of different slopes, Cheng and Howlett [12] derived the optimal driving regime corresponding to different slopes. To establish a model that adapts to any slope and speed limit, Liu and Golovitcher [13] divided the cruising into partial

traction cruising and partial braking cruising and provided the necessary conditions for the existence of the optimal switching points between two regimes. Furthermore, Albrecht and Howlett et al. [14, 15] provided a calculation formula for the optimal regime switching point on a slope, while ensuring the minimum local energy. The results from the analytic method are accurate but not suitable for solving the problems with many regime switching points. The models established to solve such a problem are often oversimplified to the point where there is a gap between theoretical analysis and practical applications.

To avoid the impact of the simplification of the actual solution, the simulation methods based on the actual situation of the railway line are often used. For example, Mao et al. [16] proposed the target-speed-control method, which allows the train to run within the preset target speed range and determines the change of driving regimes according to energy efficiency. Feng [17] analyzed the traction energy cost and transport operation time of trains at different target speeds through computer-aided simulation. However, it is difficult for such method to determine the accurate target speed. Moreover, the solutions from such simulation are rough, and the results cannot be guaranteed to be the global optimal solution.

Due to the advantages of heuristic search methods such as genetic algorithm [18–21], ant colony algorithm [18, 22–24], simulated annealing algorithm [3, 25, 26], and particle swarm optimization algorithm [27, 28] in solving nonlinear problems, they have become mainstream tools in current research on train energy-efficient optimization. Various heuristic search methods are combined into different energy-efficient models to obtain the optimal combination of driving regimes.

Two typical strategies are usually considered in the process of establishing a train energy-efficient driving model when the heuristic search methods are utilized.

One strategy is to preset a relatively fixed sequence of driving regimes, and then on it the switching points between driving regimes are searched and a combination of driving regimes containing the optimal transition point is obtained. Wong and Ho [29] dynamically allocated the number of coasting switching points, and the locations of these points are searched with the genetic algorithm. Similarly, Sandidzadeh and Alai [30] determined switching points of different regimes in a continuous domain with genetic algorithm and ant colony optimization. To find the optimal traction utilization coefficient, braking utilization coefficient, cruise position, coasting position, and braking position, He et al. [31] proposed a simulation model with an improved differential evolution algorithm. However, such models usually can only deal with a relatively fixed regime sequence, and they are not suitable for those railway lines whose gradients, slope lengths, and speed limits are kept with a constant change.

Another strategy is to directly construct a speed curve to serve as an auxiliary tool to form an optimal driving regime combination. The "speed code" model is known for constructing speed curve with a lattice composed of discrete operating intervals and speeds, and then the speed change points of the curve are obtained through the optimization search. To construct the speed curve, Lu et al. [18] applied ant colony algorithm, genetic algorithm, and dynamic programming to search the speed change points in different lattices on the "speed code" model. Zhan et al. [32] formulated the detailed train speed profile between two stations as a multiple-phase optimal control model, which is solved using a pseudo-spectral method. He et al. [33] optimized the end speed of each discrete subsegment to construct speed curve based on an improved chicken swarm optimization algorithm, considering both train energy consumption and regenerative braking energy. However, in the "speed code" model, the gradient of the connecting line between adjacent preset speed points may not reach the maximum dynamic characteristics of the train (maximum traction or maximum braking), and thus the optimal regimes derived from Pontryagin's maximum principle may not be fully utilized. Furthermore, even if a speed profile has been generated, the train will still not be able to track the profile accurately under the driving regimes determined from it since there are speed errors [34, 35].

A railroad line usually involves some complex conditions, such as many different slopes with different gradients and lengths, and these slopes are superimposed with different horizontal curves, as well as different speed limits on different gradients. The established model using the above two strategies faces difficulties in adapting to such complex conditions. To cope with such conditions mentioned above, we divide the train running section into many micro-subsegments. Because the line conditions in such subsegments are unchanged, the choice of driving regimes becomes easier. Hence, with the micro-subsegments, we propose a novel optimal driving strategy decision-making (ODSD) model for minimizing energy consumption. In this model, the required driving regimes rather than the speed curve can be directly obtained as a driving strategy is directly represented by a combination of driving regimes, avoiding the speed tracking error.

On the other hand, there are too many discretized subsegments in this case, so the driving regime selection problem becomes a discrete optimization problem for obtaining a driving regime combination. For solving discrete optimization problems, ant colony algorithm [36] and its various variants have been used naturally and widely in engineering problem [37–42]. An important component of this algorithm is a record of pheromone trails that reflect colonies' experiences with previously constructed solutions. Although the algorithm has the robustness in performance, it has also some inherent defects in updating pheromone trails [43]. Traditionally, there are two different strategies for updating pheromone trails: global-best and iteration-best. However, the former may cause a too-fast convergence of the algorithm toward some suboptimal solutions, while the latter may sometimes converge too slowly and lack focus. To improve the performance of ant colony algorithm, many scholars have studied the methods of updating pheromone trails. Afek et al. [44] presented the minimum number of pheromones necessary for a colony of ants to find a food source. Acharya et al. [45] introduced an exponentially

increasing pheromone deposition approach by artificial ants. For updating pheromone value, Myszkowski et al. [46] selected the worst or best ant found solution in a given iteration and updated the pheromone value by the worst. Ivkovic et al. [47] analyzed the effect of different pheromone trail reinforcement strategies and confirmed that numerically adjustable strategies can significantly improve algorithmic performance.

A pheromone updating strategy depends on the type of problem, heuristic information, conditions, and parameters [47, 48]. For the driving regime selection problem, the heuristic effect of the difference between the best solutions of two adjacent iterations is exploited to update pheromone for improving computing performance of ant colony system (ACS algorithm) to deal with many subsegments in the model.

The main contributions of this study are as follows:

(1) A novel flexible ODSD model is established, which is not dependent on an unfixed sequence of driving regimes. This model divides the railway section into multiple micro equal time subsegments, and finally the energy-efficient driving regime in each subsegment can be obtained directly, which adapts to complex railway line conditions and avoids the speed tracking error since there is no need to generate a speed curve.

(2) An improved ACS algorithm (named ACSd algorithm) is presented. Due to the increasing solving difficulty from many discretized subsegments, the difference edge strategy is introduced into the ACS algorithm to improve its global pheromone updating rule, which can avoid premature convergence and further improve the optimization performance.

(3) For achieving balance of the average speed and energy saving on the selection of driving regimes, heuristic information is presented and introduced into the ACSd framework in solving the ODSD model.

The remaining of this paper is as follows. The section "Optimal Train Driving Strategy Decision-Making Model" constructs the ODSD model with equal time-division pattern. The section "Solving the ODSD Model with ACSd" explains the proposed the ACSd algorithm and combines it with the ODSD model. The section "Experiments on TSP with the ACSd Algorithm" tests and analyzes the performance of ACSd in solving TSP. The section "Experiment on the ODSD Model with ACSd" applies the ODSD model and the ACSd algorithm to a case and gives the experimental results. Finally, the section "Conclusions" summarizes the main findings.

2. Optimal Train Driving Strategy Decision-Making Model

2.1. Mathematical Model Formulation. The shorter the total running time of a train on the same railway line, the greater the energy consumption [49]. This paper considers a scenario of timing requirements, i.e., the train runs with a fixed running time (T) between two adjacent stations. In the given and fixed running time, the train passes the railway section with length L from the center O of a station (initial speed $v_0 = 0$) to the center D of another station (final speed $v_N = v(T) = 0$).

The objective function of the ODSD model for the energy-efficient train is the total trip energy consumption. Since the maximum energy consumption of train traction systems can account for 85% of the total energy consumption [43], we aim to minimize the energy consumption of the train traction system. Therefore, the basic model for the energy consumption to be minimized is the work done by the traction power [3, 50]:

$$\begin{aligned} E = \min & \int_0^T u^+(t)v(t)dt \\ & \dot{x}(t) = v(t); \\ & \dot{v}(t) = u(t) - r(v(t)); \\ \text{s.t} \quad & x(0) = 0, x(T) = L, v(0) = 0, v(T) = 0; \\ & v(t) \geq 0, u(t) \in \left[-u_{\min}, u_{\max}(v(t))\right], \end{aligned} \tag{1}$$

where $v(t)$ is the speed of train; $x(t)$ is the distance traveled over time; $\dot{v}(t)$ is the derivative of speed to time; $r(v(t))$ is the resistance experienced by a unit mass train traveling at speed $v(t)$; and $u(t)$ is the traction force or braking force, when $u(t) \geq 0$ is the traction force, and when $u(t) < 0$ is the braking force; the integral only considers the positive traction force, i.e., $u^+(t) = \max(u(t), 0)$.

2.2. Equal Time-Division Pattern. Our aim is to find out a driving strategy that meets the above conditions and minimizes the energy consumption of the train. This strategy is composed of a driving regime sequence and its switching points.

For a free selection of driving regimes instead of a fixed sequence, we divide the total running time into N micro equal time subsegments (Figure 1), represented by SE_i ($i = 1, 2, \ldots, N$).

We name the structure in Figure 1 the equal time-division pattern. The total energy is the sum of the energy of each subsegment according to Figure 1. With this discrete time subsegment SE_i, equation (1) is replaced by

$$E = \min \sum_{i=1}^{N} E_i, \tag{2}$$

where $E_i = u^+(t)v(t)\Delta t$, which is the energy consumption of the train running in SE_i. The consumed energy in SE_i depends on traction force under the railway line conditions of SE_i. Besides, the regime r_i, the inlet speed v_{i-1} (arriving at SE_i), and the time step size Δt make a difference in the amount of energy consumed in SE_i.

Let the equal time subsegment size $\Delta t = T/N$. Thus, the running distance of the train in segment SE_i is as follows:

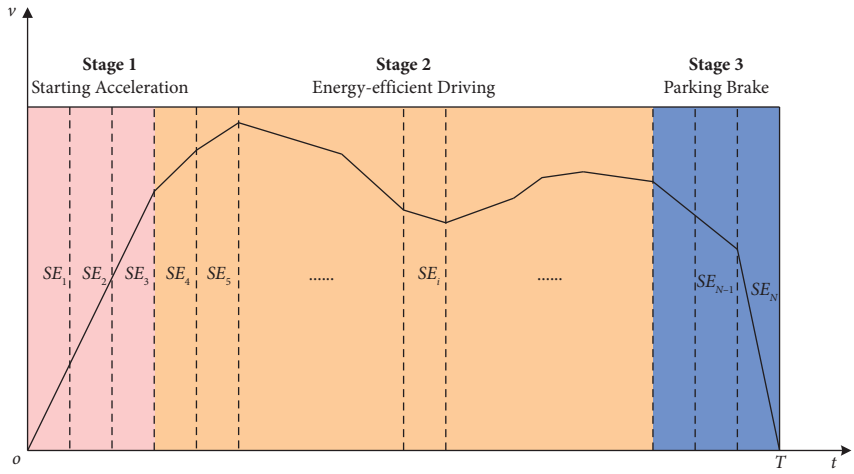

Figure 1: Equal time-division pattern.

$$S_i = v_{i-1} \cdot \Delta t + \frac{1}{2} a \Delta t^2,$$
$$a = \frac{F_t - F_b}{m} - R - R_g, \tag{3}$$

where a is the acceleration and v_{i-1} is the inlet speed of SE_i. m denotes the mass of a train, F_t is the traction force, F_b is the braking force, and R is the rolling resistance. R_g is the resistance caused by the slope, and $R_g = g \cdot \sin\theta$, where θ is the gradient of slope and g is gravitational acceleration. Δt is the equal time span corresponding to each subsegment, and there is always a certain amount of coasting time for the conversion from traction to braking, and vice versa. Consequently, the equal time subsegment size depends on the least regime-conversion time.

Thus, the sum of the running distance S_i in each equal time subsegment is equal to the total trip length L:

$$L = \sum_{i=1}^{N} S_i. \tag{4}$$

The exit speed of SE_i (expressed as $v_{i,\mathrm{exit}}$) is equal to the inlet speed of SE_{i+1} (expressed as $v_{i+1,\mathrm{inlet}}$):

$$v_{i,\mathrm{exit}} = v_{i+1,\mathrm{inlet}}. \tag{5}$$

The speed in any subsegment should not exceed the speed limit $v_{\max,i}$ required by SE_i:

$$0 \le v_i \le v_{\max,i}. \tag{6}$$

The energy consumption of trains is controlled by the regimes and the line conditions (the gradient, slope length, speed limit, etc.). In a subsegment, the gradient, slope length, and speed limit are constants, but the regime is the only decision variable in the energy consumption function. That is to say, in a subsegment, the selection of regimes has nothing to do with changing line conditions. Therefore, with these subsegments, the equal time-division pattern offers the opportunity to determine the driving strategy for a variety of line conditions, i.e., the advantage of the pattern is that the ODSD model can adapt to the changes of line conditions. Furthermore, the pattern is provided to the ACSd algorithm for regime selections: select only one regime in each subsegment and patch each one sequentially into a combination of regimes, i.e., a regime strategy (or a solution).

3. Solving the ODSD Model with ACSd

In this section, we propose an improved ant colony system for the resolution of the ODSD model, which mainly involves the regime representation, the regime choice rule, and the pheromone update rule.

3.1. The Regime Representation. The train operation from one station to the next involves three stages.

Stage 1. Starting acceleration: traction regime is always adopted, and especially, the maximum traction force should be used to speed up in a short time.

Stage 2. Energy-efficient driving stage: a combination of maximum traction and coasting is usually adopted at this stage.

Stage 3. Parking brake stage: coasting and maximum braking are adopted successively, so that the speed is zero when the train reaches the station center.

From three stages, the choice on regimes at Stage 1 and Stage 3 is clear. As shown in Figure 1, Stage 1 takes the traction regime, and Stage 3 takes the coasting regime and then braking regime, which are usually certain. There is no need for regime decisions at these two stages. By contrast, regime decisions mainly happen at Stage 2, where there are three regimes provided for decision making (see Figure 2). According to the optimal train control theory, the optimal driving regime consists of maximum acceleration, cruising, coasting, and maximum braking. However, for urban rail trains with short travel distance (generally less than 5,000 m), the cruising regime is not generally contained [51]. Besides, according to the study, driving strategies with or without cruise regime have advantages over each under

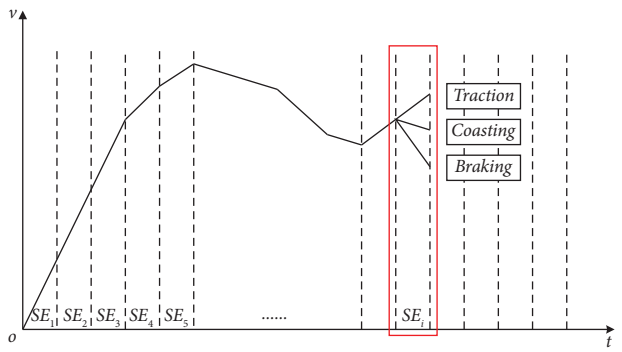

FIGURE 2: Choosing one of the three driving regimes in a sub segment.

different line environments and trainloads [31]. Therefore, to reduce the complexity of train operation optimization, we consider only maximum traction, coasting, and maximum braking in this regime decision but ignore cruising.

Finding out an optimal regime sequence is the aim of solving the ODSD model for minimizing energy consumption. A regime sequence is an orderly combination of regimes, and it is here regarded as the route of an ant when we use the ACSd algorithm to solve the model.

Let $r_i\,(r_i \in I = \{-1, 0, 1\})$ represent one of the three driving regimes of SE_i, where the set I includes three regimes: maximum traction, coasting, and maximum braking, and they are, respectively, denoted as 1, 0, and −1.

In Figure 3, the edge $R_{i,u}\,(i = 1, 2, \ldots, N; u = 1, 2, 3)$ represents the regime r_i of SE_i. There are three edges in each subsegment (SE_i) of the pattern, and they represent the three possible regimes in a subsegment. An ant selects one of the three edges which means that a driving regime is selected, and the successive selected regimes in each subsegment would construct a route.

The decision-making process is embedded in the ACSd-based framework, where the driving regimes are selected in each subsegment. A route is composed of a sequence of edges from the start to the end. In other words, a route, as a combination of the regimes on each SE_i, is a feasible solution for the energy minimization problem. A route is constructed with an ant passing through each SE_i.

The candidate set $\mathrm{allowed}_i\,(\mathrm{allowed}_i \subseteq I)$ is a component of the ACSd algorithm, which stores the possible selected regimes of the subsegment SE_i as follows:

$$r_i \in \mathrm{allowed}_i = \begin{cases} \{1, 2\}, & \text{if } r_{i-1} = 1, \text{when } \mathrm{SE}_i \text{ at Stage 1;} \\ \left.\begin{array}{ll} \{1, 2\}, & \text{if } r_{i-1} = 1, \\ \{1, 2, 3\}, & \text{if } r_{i-1} = 2, \\ \{2, 3\}, & \text{if } r_{i-1} = 3, \end{array}\right\} & \text{when } \mathrm{SE}_i \text{ at Stage 2;} \\ \left.\begin{array}{ll} \{2, 3\}, & \text{if } r_{i-1} = 2, \\ \{3\}, & \text{if } r_{i-1} = 3, \end{array}\right\} & \text{when } SE_i \text{ at Stage 3.} \end{cases} \tag{7}$$

$\mathrm{allowed}_i$ considers the continuity of the regime in adjacent subsegments, and there is always a coasting regime in-between traction and braking for their conversion.

3.2. The Regime Choice Rule in Equal Time Subsegments. In the process of solution construction, an ant selects a regime in each equal time subsegment (SE_i) with probability, i.e., the state transition rule is of ACS is used when selecting regimes. The rule is originated from heuristic information and pheromone level of the regime in the route graph.

In the ACS, an ant uses the transition rule (random proportional rule) to make probabilistic choices for its next node. Similarly, in the ODSD model, an ant iteratively chooses one regime r_i (i.e., the edge u of the ACS algorithm) in $\mathrm{allowed}_i\,(i = 1, 2, \ldots, N)$ according to the following formula:

$$r_i = \begin{cases} \underset{u \in \mathrm{allowed}_i}{\operatorname{argmax}} \left\{ [\tau_{iu}] \cdot [\eta_{iu}]^{\beta} \right\}, & \text{if } \zeta \le \zeta_0, \\ U, & \text{otherwize,} \end{cases} \tag{8}$$

where ζ is a random number uniformly distributed in [0, 1], ζ_0 is a parameter $(0 \le \zeta_0 \le 1)$, and U is a selected regime produced by roulette wheel selection for ant k according to the following:

$$p_{i,u}^{k} = \begin{cases} \dfrac{[\tau_{iu}] \cdot [\eta_{iu}]^{\beta}}{\sum_{u \in \mathrm{allowed}_i} [\tau_{iu}] \cdot [\eta_{iu}]^{\beta}}, & \text{if } u \in \mathrm{allowed}_i, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where τ_{iu} is the accumulating pheromone (i.e., the level of pheromone deposited) on edge $R_{i,u}^{k}$; η_{iu} is the heuristic information of edge $R_{i,u}^{k}$; β is the parameter controlling the relative importance of pheromone τ_{iu} versus heuristic information $\eta_{iu}\,(\beta > 0)$; and τ_{iu} and η_{iu} are considered the key for solving the ODSD model, and they are discussed in the next two subsections.

3.3. Heuristic Information. Heuristic information on the edge $R_{i,u}^{k}$ works for ants to decide which regime to be selected. Reducing the energy consumption and keeping the appropriate speed are the main considerations of the heuristic information for the ODSD model. Thus, we introduce two heuristic factors η_1 and η_2 to express the heuristic information for the edge section.

3.3.1. Energy-Efficient Heuristic Factor. The energy-efficient heuristic factor η_1 reflects the heuristic and guiding effect on the energy consumption when selecting a regime in SE_i $(i = 1, 2, \ldots, N)$. We first thought that $1/E_{ir}$ may be used as the heuristic factor. However, it would malfunction because the energy consumption of the coasting is zero. Thus, a modification is considered as follows:

$$\eta_1 = \frac{1}{\lambda + E_{ir}}, \tag{10}$$

where λ is a regulator for alleviating the excessive influence of some regimes on the regime decision, and let $\lambda = \gamma \cdot (E_{i1} + E_{i2} + E_{i3})$, where γ is the amplification factor (let $\gamma = 10$ from our experiments) and E_{i1}, E_{i2}, and E_{i3}

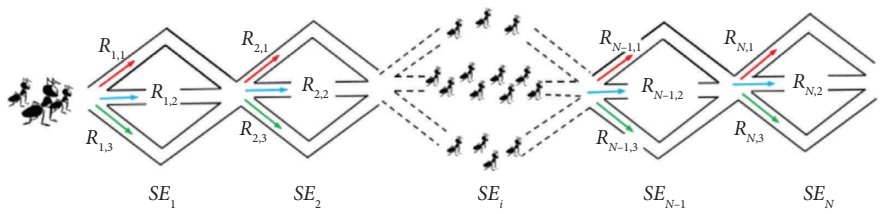

Figure 3: The route graph of the ants.

represent the energy consumption of using traction, coasting, and braking in SE_i, respectively.

3.3.2. Speed Heuristic Factor. Although the energy-efficient heuristic factor η_1 is the main heuristic factor for reducing the energy consumption, the train speed will fall too low if this factor exerts excessive impact. This is because low consumption requires coasting. Nonetheless, it brings about low-speed operation and is prone to violate the timing requirements.

To overcome this problem, the reference speed of the slope is hereby introduced into the heuristic information as another factor (η_2) to guide the decision of regimes. The reference speed of slope q ($\overline{v}_q, q = 1, 2, \ldots, Q$) refers to a rough speed, as a reference baseline for the running speed (different slopes have different reference speeds).

The reference speed $\overline{v}_q$ is represented by the ratio of the slope length L_q to the reference running time $t_{\mathrm{ref},q}$ ($q = 1, 2, \ldots, Q$) on slope q:

$$\overline{v}_q = \frac{L_q}{t_{\mathrm{ref},q}}, \tag{11}$$

where $t_{\mathrm{ref},q}$ is the running time on slop q of the current best solution (i.e., current best regime combination).

To provide the guiding effect on the running speed and avoid overusing coasting, $\overline{v}_q$ is introduced into the heuristic factor η_2:

$$\eta_2 = \frac{1}{\left|v_{ir} - \overline{v}_q\right| + \varepsilon}, \tag{12}$$

where $\overline{v}_q$ is the reference speed on slope q, which can be calculated by equation (11), and v_{ir} is the speed in SE_i under regime r. $|v_{ir} - \overline{v}_q|$ reflects the difference between the current speed and the reference speed. The closer v_{ir} is to $\overline{v}_q$, the larger the denominator is. This indicates a higher probability of regime r being selected. A small number ε is added into the denominator (for example, let $\varepsilon = 0.001$) to avoid that the denominator is zero. In addition, the smaller the deviation, the greater the visibility of regime for an ant. In other words, the introduction of the reference speed $\overline{v}_q$ provides a trade-off between timing requirement and energy-efficient consumption.

With the above process, we can get η_2; however, at the first iteration, because no solution is constructed at this time, we cannot get $t_{\mathrm{ref},q}$. For this case, $t_{\mathrm{ref},q}$ is estimated by equation (13). Suppose that $t_{\mathrm{ref},q}$ at the first iteration is directly proportional to the time cost on slope q at the allowed speed $v_{\max,q}$:

$$\begin{aligned} \frac{t_{\mathrm{ref},q}}{L_q/v_{\max,q}} &= \frac{t_{\mathrm{ref},q+1}}{L_{q+1}/v_{\max,q+1}} \\ &= \ldots = \frac{t_{\mathrm{ref},Q}}{L_Q/v_{\max,Q}}, \end{aligned} \tag{13}$$

where L_q is the length of slope q ($q = 1,2, \ldots, Q$).

For solving $t_{\mathrm{ref},q}, t_{\mathrm{ref},q+1}, \ldots, t_{ref,Q}$ of equation (13), we add the following formula to support it. Let SE_p be the last subsegment on slope $q-1$, and hence $\sum_{j=1}^{q-1} L_j = \sum_{i=1}^{p} S_i$.

According to $t_i = i \cdot \Delta t$ ($i = 1,2, \ldots, N$), we obtain

$$\sum_{j=q}^{Q} t_{\mathrm{ref},j} = T - \sum_{i=1}^{p} t_i, \tag{14}$$

where $\sum_{i=1}^{p} t_i$ is the time that the train has run on reaching the end of SE_p. With equations (13) and (14), we can also attain $\overline{v}_q$ at the first iteration by equation (11).

Energy-efficient heuristic factor η_1 and speed heuristic factor η_2, from two different views, provide heuristic information for an ant to choose a regime. For a synthetic effect, let η_{ir} be a heuristic function in equations (10) and (12):

$$\eta_{ir} = \eta_1 \times \eta_2. \tag{15}$$

3.4. The Pheromone Update Rule. The local pheromone updating rule of the ACS algorithm can usually avoid the accumulation of pheromone on an edge, so that ants can find other new edges, whereas the global pheromone updating rule can result in those edges of the best route to be selected with high probability. However, there may be missing heuristic information in the difference edges between the best routes of two adjacent iterations. Therefore, based on the global pheromone updating rule, we propose the difference edge strategy.

3.4.1. The Local Pheromone Updating Rule. The local pheromone updating rule means that whenever an ant moves from one subsegment to another, the pheromone on the edge is eliminated/evaporated, i.e., the pheromone of the edge (regime r_i in SE_i) is calculated by equation (16). The local adjustment and evaporation of pheromone can reduce

aggregation of pheromone and promote ants to find new edges in a wider neighborhood of the previous tour.

$$\tau_{iu} = (1 - \xi) \cdot \tau_{\mathrm{iu}} + \xi \cdot \tau_0, \tag{16}$$

where $\xi \in [0, 1]$ is the evaporation rate of the local pheromone and τ_0 is the initial pheromone level on the route.

3.4.2. The Global Pheromone Updating Rule. After all ants have reached the end at iteration s ($s = 1, 2, \ldots,$ ite_max), the global pheromone updating rule updates the pheromone of the best route of the current iteration (called the iteration-best route), and the pheromone of each edge is updated by the following equation:

$$\tau_{iu} = (1 - \rho) \cdot \tau_{\mathrm{iu}} + \Delta\tau_{\mathrm{iu}}^{\mathrm{best}}, \tag{17}$$

$$\Delta\tau_{\mathrm{iu}}^{\mathrm{best}} = \begin{cases} \left(E_{I\_\mathrm{Route}_s}\right)^{-1}, & \text{if } R_{i,u} \in I\_\mathrm{Route}_s, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

In equation (17), $\rho \in [0, 1]$ is the evaporation rate of the global pheromone and $\Delta\tau_{\mathrm{iu}}^{\mathrm{best}}$ is the pheromone increment on the iteration best edge. In equation (18), I_Route_s represents the iteration-best route at iteration s and $E_{I\_\mathrm{Route}_s}$ is the minimum energy consumption under the regimes corresponding to the iteration-best route.

3.4.3. Pheromone Updating Strategy on Difference Edges. With the increase of the complexity of optimization problems, the original ACS algorithm should be revised to avoid the premature convergence and improve the solution quality.

During the pheromone update, the pheromone of the original ACS algorithm is deposited on the best route [39]. With the increase of the iterations, more pheromones would accumulate on the best route, which would lead to search stagnation. However, the difference between two successive iteration-best solutions may suggest a new heuristic [52]. Hence, we propose a new pheromone updating strategy for the ACS algorithm and introduce it into the ODSD model to improve the calculation performance for the optimal driving strategy.

During the iteration, the iteration-best route works in a state of change. Comparing the two iteration-best routes at iteration s and iteration $s-1$, those different edges between them are defined as difference edges. The difference edges come from the comparison between the current iteration-best route (represented by I_Route_s) and the preceding iteration-best route (represented by I_Route_{s-1}). For exploiting difference information, we add reinforcement of pheromone to these difference edges in addition to the pheromone changes in equations (16) and (17).

For example, suppose that

$$\begin{aligned} I\_\mathrm{Route}_{s-1} &= \{R_{1-2}, R_{2-3}, R_{3-4}, R_{4-5}\}, \\ I\_\mathrm{Route}_s &= \{R_{1-2}, R_{2-3}, R_{3-5}, R_{5-4}\}. \end{aligned} \tag{19}$$

We add the difference edges to the set S_{Dif}:

$$S_{\mathrm{Dif}} = \{R_{3-5}, R_{5-4}, R_{3-4}, R_{4-5}\}, \tag{20}$$

where R_{A-B} represents the edge linking node A to B.

Thus, the new ACS (ACS with difference edges, i.e., ACSd) performs the additional pheromone updating by increasing the extra pheromone $\Delta\tau_{\mathrm{iu}}^{\mathrm{Nbest}}$ to each edge in S_{Dif}:

$$\begin{aligned} \tau_{iu} &= \tau_{iu} + \Delta\tau_{\mathrm{iu}}^{\mathrm{Nbest}}, \\ \Delta\tau_{\mathrm{iu}}^{\mathrm{Nbest}} &= \begin{cases} \dfrac{1}{E_{I\_\mathrm{Route}_s}}, & \text{if } R_{i,u} \in S_{\mathrm{Dif}}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{21}$$

The difference edge strategy is illustrated in Figure 4. We assume that the initial pheromone on each edge is zero, and the pheromone increment is 1 on the edge once an ant passes through it.

Figure 4(a) shows the iteration-best route of iteration $s-1$, and according to the global pheromone updating rule (see equation (17)), the pheromone amount increases to 1 on each edge. The red route in Figure 4(b) is the iteration-best route of iteration s. For the two routes, many edges are the same. This is because under the induction of pheromone aggregation, ants always prefer to pass through those edges with high pheromone density; thus, the ants walk over some edges again, which results in many overlapped edges between the two routes. However, some edges do not overlap, and these nonoverlapped edges are the very difference edges. Thus, the pheromone increases to 2 for the same edges, whereas the pheromone increases to 1 for the difference edges. Obviously, the pheromone of difference edges is less than those same edges.

In fact, the two routes are iteration-best routes, so they are heuristic for further search, and the lower pheromone concentration on difference edges may reduce the heuristic effect. Therefore, as shown in Figure 4(c), pheromone is added to the difference edges. Adding pheromone on the difference edges may intensify the heuristic effect, as additional pheromone can provide guidance for exploiting search space. This may help to avoid premature convergence and provide better solution compared with the original ACS algorithm.

The new pheromone updating strategy for difference edges is different from ACS. For the sake of distinction, we call the algorithm ACSd. The experiment in Section 4 shows that ACSd has significant effect and performs better than ACS.

3.5. Constraint Violation and Repair. There are some constraints in the problem of optimal energy-efficient train driving strategy, such as speed limits and reaching next station at the stipulated time.

The penalty method may be the most common technique to solve constrained optimization problems. However, a large penalty value would result in premature convergence, while a small penalty would produce many infeasible solutions [53]. Hence, we take a repair procedure for

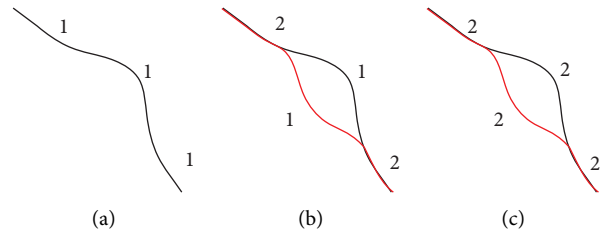

Figure 4: Pheromone change on difference edges.

constraints to replace the penalty method. This procedure attempts to fix infeasible solutions by considering the ODSD model itself.

3.5.1. Speed Constraints. During the route construction, at some positions such as steep downgrade, the speed of the train would reach the limit. To reduce the speed, the traction regimes of some subsegments should be substituted for braking or coasting, and the substitution is conducted in a trial-and-error manner. For example, assuming the speed violation happens in SE_i, replace the traction regime in SE_{i-1} with coasting regime backward and then recalculate the speed in SE_i. Then, check if the speed limit is exceeded; if exceeded, change the regime with braking regime.

If the speed still exceeds the speed limit, the above process will be repeated ahead of subsegment SE_{i-1} (i.e., SE_{i-2}).

3.5.2. Distance Constraints. ACSd selects driving regimes in each subsegment SE_i ($i = 1, 2, \ldots, N$), on which the time is equal, but the length S_i is not equal. S_i varies with the different regimes of SE_i, and $\sum_{i=1}^{N} S_i$ is the sum length of all subsegments, theoretically equal to the total trip distance L. Equation (22) represents this relationship within the allowable error δ:

$$\Delta S = \left| L - \sum_{i=1}^{N} S_i \right| \leq \delta. \tag{22}$$

Since the length of each subsegment varies with different regimes, we change the regimes used in some subsegments to reduce or increase the subsegment length when equation (22) is not satisfied.

According to the Davis equation [54], for the train in traction, energy consumption increases with the increase of speed on the uphill slope. Thus, for reducing the energy consumption, we modify the regimes in subsegments that produce the maximum speed.

Accordingly, if $\sum_{i=1}^{N} S_i > L$, the traction regimes of some subsegments should be modified to coasting regime for reducing $\sum_{i=1}^{N} S_i$; if $\sum_{i=1}^{N} S_i < L$, some selected coasting regimes should be substituted to traction regimes for increasing $\sum_{i=1}^{N} S_i$.

The case of $\sum_{i=1}^{N} S_i > L$ is used as an example to explain the above process as follows. In Figure 5(a), SE_5 is a switching subsegment between two regimes (i.e., before point B is traction, and after point B is coasting). Moreover, SE_5 is a subsegment immediately preceding point B with the highest speed in the uphill process. We substitute the coasting for the traction in SE_5 and recalculate the sum of running distance and check whether it meets (22). During the process of repeating the above steps, subsegments may be too long for meeting the small δ. In this case, the subsegment is further divided into subsegments (see Figure 5(b)). Replace traction with coasting in the subsegments one by one until the results meet constraint (22).

3.6. The Algorithm Framework for the ODSD Model. Algorithm 1 lists the process of constructing a feasible solution, which is a route of an ant from the start to the end. A route is composed of N regimes (edges). Ant k at iteration s selects a regime (edge) in SE_i using the state transition rule. For using the rule, the heuristic information including energy-efficient factor and speed factor is calculated in advance. In the process, the regime is replaced backwards when the speed exceeds the limits, and the distance constraint violation is checked when the ant reaches the terminal.

Algorithm 2 shows the process of ACSd in solving the ODSD model. The step of constructing a feasible solution is a necessary component. The difference edges are identified by comparing the two adjacent iteration-best routes.

4. Experiments on TSP with the ACSd Algorithm

We add the difference edge strategy to update the global pheromone in Section 3, which is a new operation on the basic structure; therefore, it should be contrasted with the basic ACS to examine the effect of the difference edge strategy.

4.1. Procedure of the ACSd Algorithm. The independent procedure of the ACSd is described as follows (suppose there are n cities to be visited).

Step 1. Initialize the parameters of the ACSd including $m, \mathrm{ite\_max}, \beta, \xi, \rho, \tau_0, \eta, q_0$.

Step 2. For m ants, select m cities for a start randomly.

Step 3. For each ant, select the next city j in $\mathrm{allowed}_i$ ($i = 1, 2, \ldots, n$) according to equations (23) and (24), where $\mathrm{allowed}_i$ is the set of cities that remain to be visited by the ant positioned on city i.

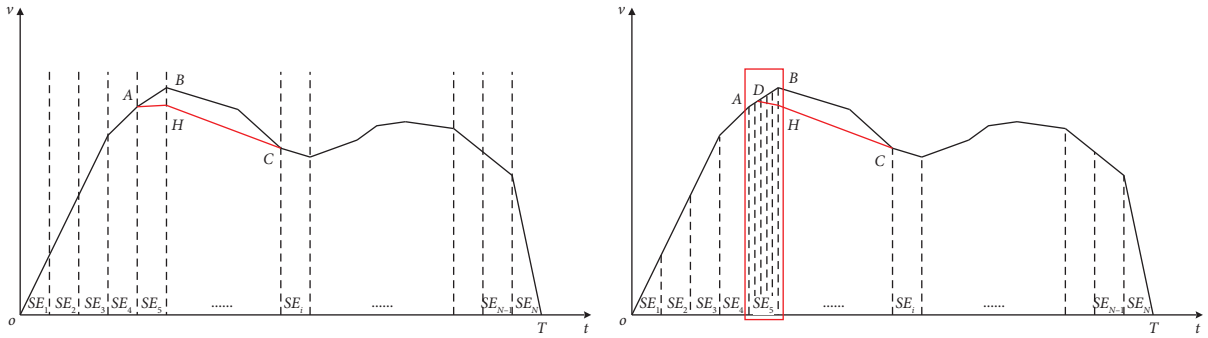

FIGURE 5: Two-stage correction process for an example when $\sum_{i=1}^{N} S_i > L$ (A-D-B-C is the original speed-time curve, and A-D-H-C is the modified one).

Given $[\tau_{iu}]$ in SE_i $(i = 1, 2, \ldots, N; r \in I)$;
(1) for $(i = 1, i < N, i++)$
(2) Generate a random number $\zeta \in [0, 1]$;
(3) Calculate v_{ir}, S_i and E_{ir} in SE_i;
(4) Calculate energy-efficient heuristic factor η_1;
(5) if $\sum_1^i S_i < \sum_1^q S_q$ then $q++$; //come into next slope
(6) Update the $\overline{v}_q$ of the current slope q by using equation (11);
(7) Calculate speed heuristic factor η_2 with $\overline{v}_q$;
(8) Calculate heuristic information $\eta_{ir} = \eta_1 \times \eta_2$;
(9) Choose a regime u_i in SE_i by using transition rule equation (21);
(10) if $v_{ir} > v_{q,\max}$ then
repairing speed violation
by regimes replacement backwards until $v_{ir} < v_{q,\max}$;
end if
(11) end for
(12) If $\Delta S = |L - \sum_{i=1}^{N} S_i| > \delta$, then repair distance constraint violation;
(13) return Route_s.

ALGORITHM 1: The process of constructing a feasible solution.

(1) Input: $t_i, T, L, L_q, v_{q,\max}, \overline{v}_q, I$;
(2) Initialize each τ_{iu} of edge $R_{i,r}$ in SE_i (i = 1, 2, . . . , N; u = −1,0,1);
(3) Set allowed_i for each SE_i;
(4) Set iteration counter s = 0;
(5) From m initial routes, choose the optimal route as BestRoute_s
(6) for $(s = 1, s < \text{ite\_max}, s++)$
(7) for each ant k do
(8) Construct a feasible solution Route_F using Algorithm 1;
(9) if $f(\text{Route}_F) < f(\text{I\_Route}_s)$
(10) $\text{I\_Route}_s \longleftarrow \text{Route}_F$
(11) end for
(12) get S_{Dif} by comparing I_Route_s with I_Route_{s-1};
(13) for each edge $R_{i,u}$ in I_Route_s do
(14) if edge $R_{i,u} \in S_{\text{Dif}}$
(15) $\tau_{iu} = (1 - \rho) \cdot \tau_{iu} + \Delta\tau_{iu}^{\text{best}} + \Delta\tau_{iu}^{\text{Nbest}}$
(16) else
(17) $\tau_{iu} = (1 - \rho) \cdot \tau_{iu} + \Delta\tau_{iu}^{\text{best}}$
(18) end if
(19) end for
(20) Output: Global – best_Route_s.

ALGORITHM 2: Solving the ODSD with ACSd.

$$j = \begin{cases} \underset{j \in \text{allowed}_i}{\text{argmax}} \left\{ \left[\tau_{ij}\right] \cdot \left[\eta_{ij}\right]^{\beta} \right\}, & \text{if } q \le q_0, \\ U, & \text{otherwize}, \end{cases} \tag{23}$$

where q_0 is a parameter $(0 < q_0 < 1)$, q is a random number uniformly distributed in [0, 1], and U is a random selected city according to the probability distribution given in the following equation:

$$p_{i,j}^{k} = \begin{cases} \dfrac{\left[\tau_{ij}\right] \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{j \in \text{allowed}_i} \left[\tau_{ij}\right] \cdot \left[\eta_{ij}\right]^{\beta}}, & \text{if } j \in \text{allowed}_i, \\ 0, & \text{otherwise}. \end{cases} \tag{24}$$

Step 4. After an ant builds its route, update the local pheromone of its tour according to the following equation:

$$\tau_{ij} = (1 - \xi) \cdot \tau_{\text{ij}}^{\text{best}} + \xi \cdot \tau_0, \tag{25}$$

where $\xi \in [0,1]$ is the evaporation rate of the local pheromone and τ_0 is the initial pheromone level on the route.

Step 5. After m ants finish their tours, iteration++.

Step 6. Update the pheromone of the iteration-best route when it is found at iteration s.

$$\begin{aligned} \tau_{ij} &= (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{\text{best}}, \\ \Delta\tau_{ij}^{\text{best}} &= \begin{cases} \left(L_{I\_\text{Route}_s}\right)^{-1}, & \text{if } (i, j) \in I\_Route_s, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{26}$$

where I_Route_s represents the edge set of iteration-best route.

Step 7. Update the pheromone of difference edges: compare the best route of the current iteration with that of its last iteration to find out difference edges and then update their pheromone.

$$\begin{aligned} \tau_{ij} &= \tau_{ij} + \Delta\tau_{ij}^{\text{Nbest}}, \\ \Delta\tau_{ij}^{\text{Nbest}} &= \begin{cases} \left(L_{I\_\text{Route}_s}\right)^{-1}, & \text{if } (i, j) \in S_{\text{Dif}}, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{27}$$

where I_Route_s represents the edge set of the iteration-best route. S_{Dif} is the difference edge set, which comes from the comparison between the two successive iteration-best routes.

Step 8. If iteration <ite_max, go to Step 3; else output the global-best route (i.e., the final solution) and terminate the run.

4.2. Difference Edge Strategy Improves the Performance of the ACSd Algorithm. TSP is one of the famous problems for testing a discrete optimization algorithm, and ant colony algorithm is usually based on this problem to test the performance; thus, to compare the performance between the new algorithm ACSd and the existing algorithm ACS, we test them with a few instances from TSPLIB standard library (https://comopt.ifi.uni-heidelberg.de/software/TSPLIB95).

Due to the stochastic nature of evolution, measuring the performance of algorithms is a challenging task. Traditionally, the arithmetic mean is used to measuring average performance of algorithms. Ivkovic et al. [55] demonstrated that the arithmetic mean was inadequate for measuring average performance based on the observed number of function evaluations, and they thought that the quantiles were more suitable for measuring average performance [56]. Therefore, we replace the average value with the median (i.e., 0.5 quantile, $Q_{0.5}$) to measure the average performance of the algorithm and use 0.1 quantile ($Q_{0.1}$) and 0.9 quantile ($Q_{0.9}$) to measure the peak and bad-case performance of the algorithm.

The parameter settings of ACS and ACSd are the same (see Table 1). We performed 20 independent runs at the iteration number NC_max = 3,000. The results include the maximum value, the median value ($Q_{0.5}$), the 0.1 quantile ($Q_{0.1}$), the 0.9 quantile ($Q_{0.9}$), the minimum value, and standard deviation (S.t.), which are shown in Table 2. Among them, the best results are highlighted in italic.

In Table 2, Opt. represents the best-known route length, and Err. represents the percentage of error between the minimum length and the best-known route length.

It can be clearly seen from this table that ACSd is significantly superior to ACS in all results except for the standard deviation of instance d493 (378.12 > 251.03). Also, from the column Err., we see that the minimum route length of the ACSd algorithm is almost close to the best-known results, though no other advanced operations are introduced.

Because the difference between the two algorithms lies in only the difference edges and other structures of the ACS are not changed, it indicates that the difference edge strategy does play a significant role in improving the performance of the ACS.

4.3. Analyzing Convergence of Difference Edges Strategy. For analyzing the difference edge strategy, Figure 6 shows the convergence curves of the instances in Table 2 for the ACS and the ACSd algorithms. It is worth noting that the vertical axis is the length of the iteration-best route, instead of the best-so-far route. Comparing the two convergence curves from the two algorithms for the same instance, we can see some convergence characteristics of the two algorithms, which may provide some special information for explaining the effect of the difference edges.

For ACS, the iteration-best value fluctuates greatly up and down with the number of iterations. On the whole, its oscillation almost has no downward trend, and the best-so-far solution is just occasionally found. In contrast, for ACSd, the iteration-best value has an obvious downward trend in spite of some fluctuations, which shows that the convergence of ACSd is faster than ACS.

Table 1: The set of the parameters.

Parameter	ACSd
Number of ants (m)	100
Weighted value of heuristic information to pheromone (β)	2
The evaporation rate of the local pheromone (ξ)	0.005
The evaporation rate of the global pheromone (ρ)	0.005
Number of iterations (ite_max)	3,000

Table 2: Comparison of the results between ACS and ACSd.

Instances	Algorithm	Opt.	Max	$Q_{0.9}$	$Q_{0.5}$	$Q_{0.1}$	Min	Err. (%)	S.t.
eil51	ACS	426	455.152	453.881	449.160	446.438	443.176	4.0	3.25
	ACSd		*429.484*	*429.032*	*428.982*	*428.982*	*428.872*	*0.7*	*0.15*
KroA100	ACS	21282	22884.7	22689.2	22479.2	22374.4	22300.1	4.8	151.12
	ACSd		*21628*	*21587.8*	*21307.4*	*21285.4*	*21285.4*	*0.0*	*122.59*
eil101	ACS	629	704.712	697.434	690.385	677.283	671.198	6.7	8.74
	ACSd		*665.174*	*655.826*	*650.095*	*644.975*	*642.866*	*2.2*	*5.33*
d198	ACS	15780	17099.30	17069.70	16818.85	16537.02	16456.50	4.3	194.23
	ACSd		*16163.30*	*16071.74*	*16009.60*	*15963.46*	*15933.20*	*1.0*	*51.28*
KroA200	ACS	29368	32305.10	32186.30	31802.30	31286.43	31209.70	6.3	329.95
	ACSd		*30098.60*	*29922.71*	*29667.55*	*29518.40*	*29469.00*	*0.3*	*160.66*
d493	ACS	35002	40688.80	40403.00	40117.25	39860.34	39687.10	13.4	*251.03*
	ACSd		*39895.80*	*39654.84*	*39286.60*	*38747.83*	*38395.60*	*9.7*	378.12

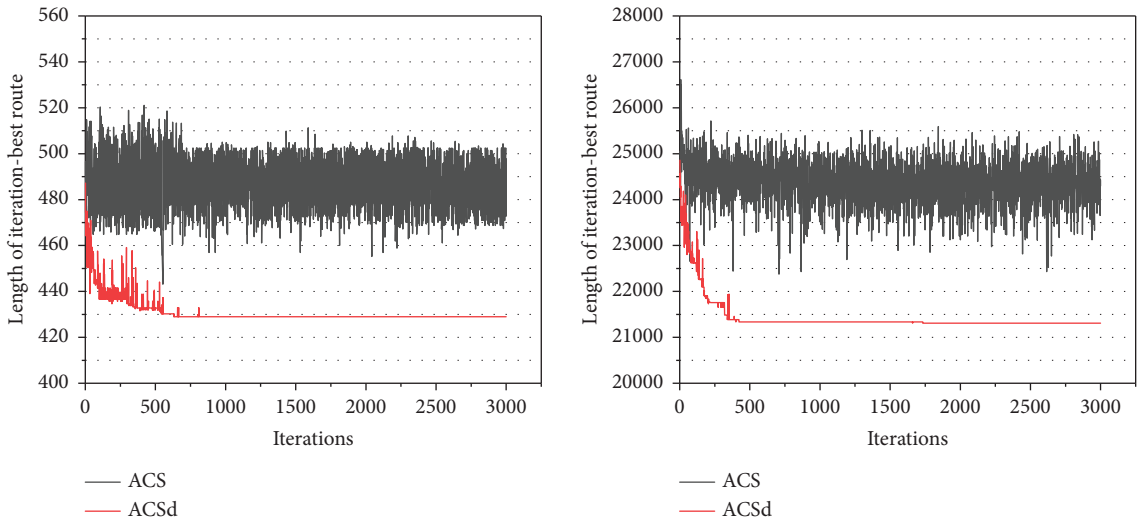

Figure 6: Continued.

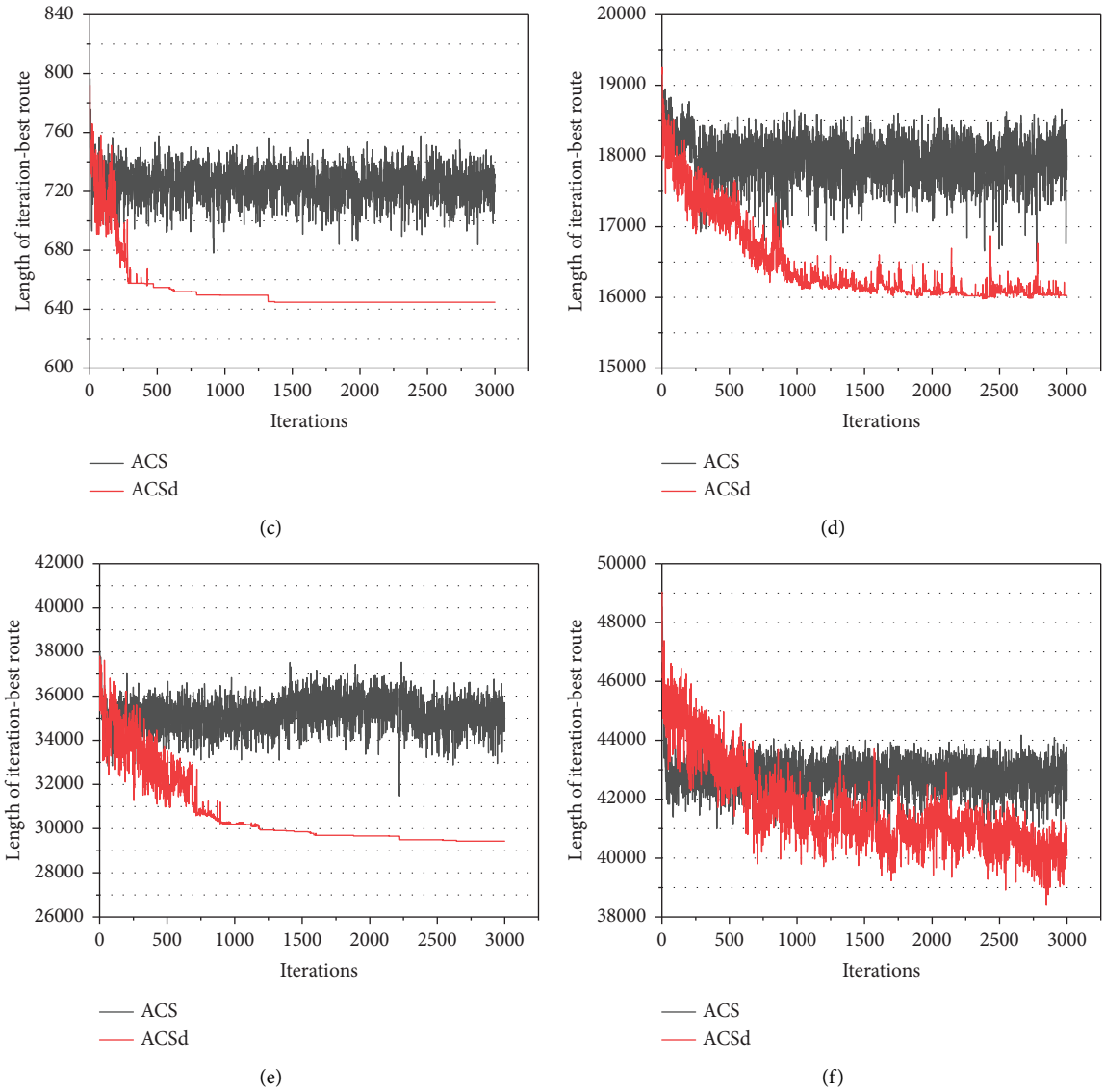

Figure 6: Convergence graph of the iteration-best route of the ACS and the ACSd algorithms. (a) eil51. (b) KroA100. (c) eil101. (d) d198. (e) KroA200. (f) d493.

The curve of ACSd is generally lower than that of ACS (i.e., the red curve is below the black curve), indicating that the search of ACSd is in a strong level. Even at the beginning phase of the search, it is easy for ACSd to find a better solution than ACS except for d493.

Therefore, we can make the following conclusion: difference edge strategy does improve the performance of the ACS algorithm. That is to say, with the information carried by the difference edge being strengthened, the difference edge strategy has an additional heuristic effect on the subsequent search, and thus ACSd has better exploration ability and faster convergence speed.

4.4. Parameter Experiments. The values of parameters in the ACSd algorithms could cause different optimal results. For the suitable parameters, the number of ants (m), the weighted value of heuristic information to pheromone (β), the evaporation rate of the local pheromone (ξ), and the evaporation rate of the global pheromone (ρ) are tested. Before the experiment, with a preliminary test, we estimate $\rho, \zeta \in [0, 0.05]$, $\beta \in [1,5]$ as the parameter interval of the ACSd algorithm, and for the number of ants, we test it in a general range ($m = 10 \sim 200$).

In the experiments, only one parameter is changed in a trial for showing the effect of this factor. The representative test cases for the parameter setting are done with 20 independent runs on a selected instance (d198), and the experimental results are shown in Figure 7.

From Figure 7, we gain the appropriate value (presented in Table 1) for the four parameters when the fitness evaluation value is the smallest.

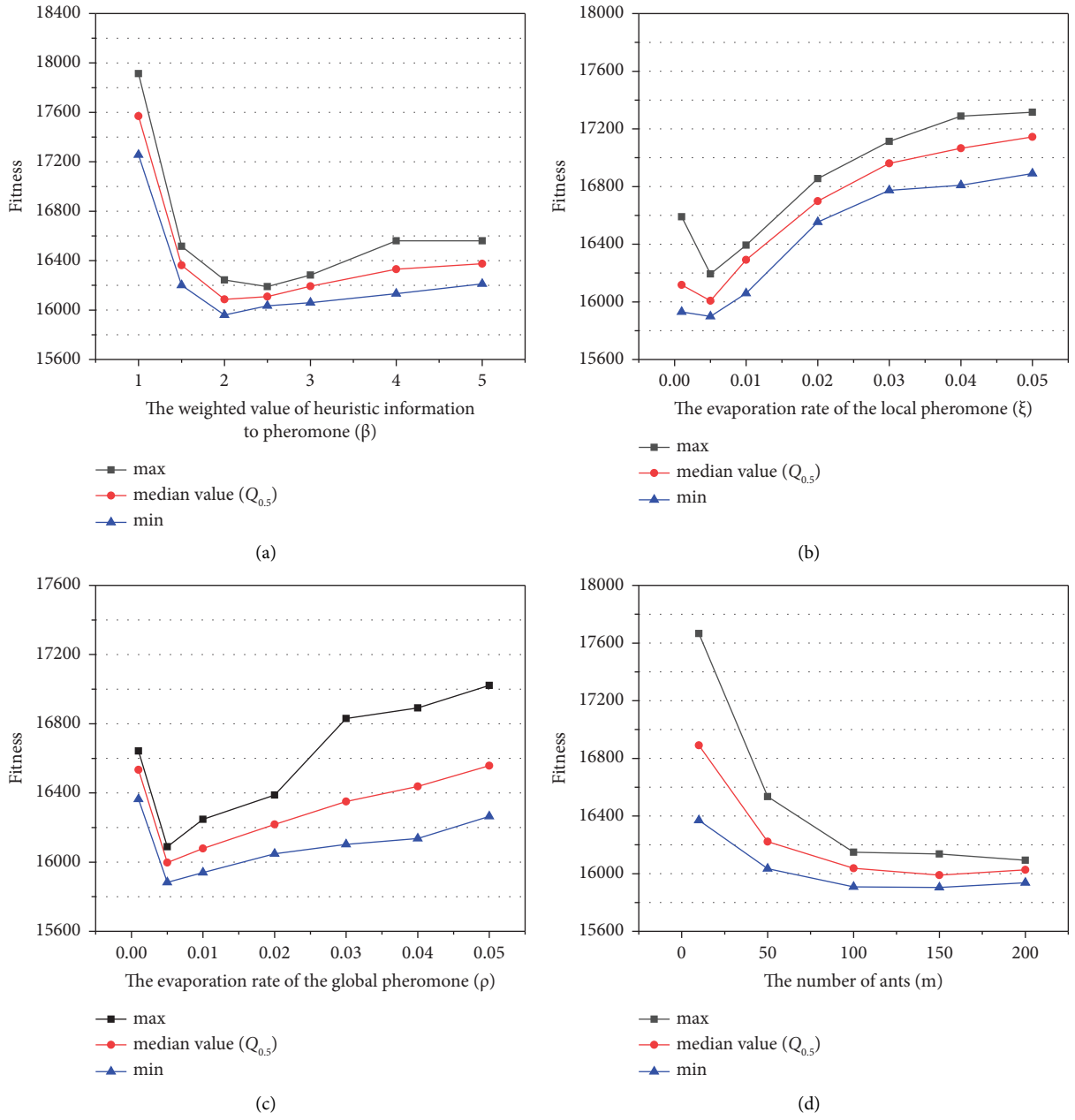

Figure 7: The test of parameter sensitivity.

5. Experiment on the ODSD Model with ACSd

The proposed ODSD model and ACSd algorithm are applied to a case from a previous study for validation [57]. Most existing methods or models for solving optimal driving strategy of energy-efficient train problem are tested or applied on different real-world instances. Considering different scenes or constraints, it is not easy for them to make a fair comparison. In our work, we used the same instance of the real-world problem for their results are available [57].

In this case, the train is required to run a total distance of 20 km and a fixed time of 25 minutes. The slope length (m) and the gradient (‰) of each slope section are shown in Figure 8. The speed limit of the train is 80 km/h, and the turnout speed limit is 45 km/h (the distance from the turnout to the end point is 1,600 m). The train is pulled by electric locomotive with the traction weight of 3,000 t, and the train conversion braking rate $\theta_h = 0.3$. According to the requirement of model discretization, the trip time is divided into 500 subsegments, i.e., the time subsegment size is set as 3 seconds in the equal time-division pattern. Additionally, the ACSd algorithm is applied in this instance, and its parameters ρ, β, ξ, and m are set according to Table 1, and the number of iterations is set to 1,000.

We aim to decide the driving regime in each time subsegment to obtain the final regime sequence and calculate the speed-time curve and speed-distance curve of the train operation to visually display the optimization results.

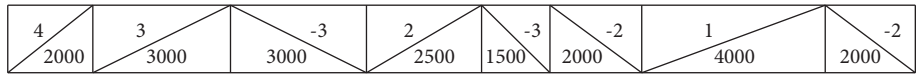

Figure 8: The slope information of the example.

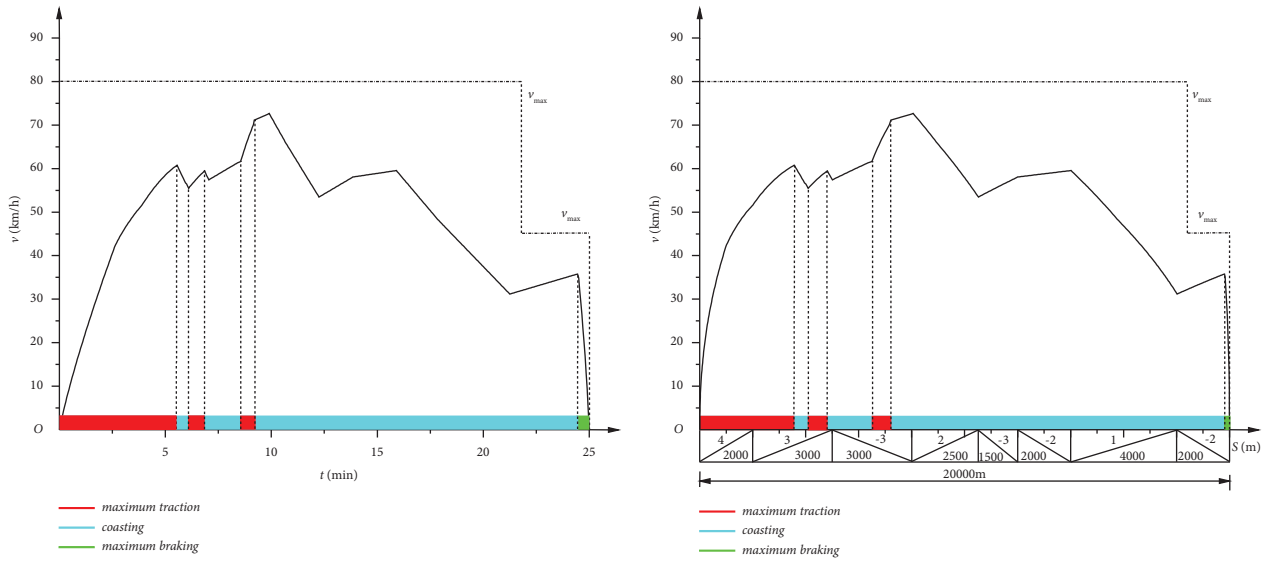

Figure 9: Analysis of simulation results of optimal driving strategy. (a) Energy-efficient speed-time curve. (b) Energy-efficient speed-distance curve.

Table 3: Comparison of results of different methods.

Model	Total time (min)	Total distance (m)	Energy consumption (kW·h)
Time-efficient	20.53	20,000.0	692.21
Target-speed control	24.99	19,999.9	652.52
ODSD model (ACSd)[1]	25.00	20,003.3	609.86
ODSD model (ACSd)[2]	25.00	19,997.4	604.90

[1,2]The maximum iteration number is 300 and 1000, respectively.

The final regime sequence is {traction, coasting, traction, coasting, traction, coasting, braking}, and their conversion of regimes occurred at $t = 5.60$ min, 6.15 min, 6.90 min, 8.60 min, 9.20 min, and 24.50 min, respectively, or at $S = 3{,}617.3$ m, 4,142.8 m, 4,865.7 m, 6,551.9 m, 7,219.0 m, and 19,835.8 m. The speed-time curve and the speed-distance curve are shown in Figures 9(a) and 9(b), respectively. The energy consumption of this strategy is 604.90 kW·h. Additionally, to check the effect on the results, the maximum number of iterations is set as 300, and its energy consumption is 609.86 kw · h. The difference between the results of two different iterations is not significant, which suggests that the convergence rate becomes small from iteration 300 to iteration 1,000.

For comparison, we consider the time-efficient driving strategy model and the target-speed control model [57]. The former is presented for minimizing running time, and the latter is a kind of energy-efficient model considering timing requirements, which allows a train to run within a preset target speed range according to energy-saving principles. Our ODSD model also involves the timing requirements; however, it provides the driving strategy that consumes the lowest energy consumption. From Table 3, compared with time-efficient model and target-speed control model, the energy consumption of our method is reduced by 13.5% and 7%, respectively.

In addition to the efficiency brought by the improved algorithm, the reason for this result may be owing to the equal time-division pattern: selecting regimes in each subsegment by the ACSd may provide more flexible driving strategies for the ODSD model. Our approach adopts heuristic process with the short time subsegment, which provides more opportunities to find better results from more regime combinations. Also, with dividing the section into small subsegments, it can avoid the influence of slope lengths, gradients, and speed limits on regime selection, and thus it is easy to adapt to different line conditions.

In contrast, the other two models require the train to run within a preset target speed range for energy-efficient concerns, and only when the speed reaches the boundary of the target speed range or is close to the change point of slope, does the regime change. Therefore, this conversion is not flexible enough to get a good solution.

6. Conclusions

This paper develops a novel ODSD model for the minimization of train energy consumption. In the ODSD model, the driving regimes can be directly selected and applied to control train operation, so as to avoid the speed tracking error. Besides, with the support of the equal time-division pattern, the model may produce the optimal strategy for a train to adapt to a wider range of railway line conditions. This pattern, which is constructed by discretizing the total running time into equal subsegments, provides a basis for selecting the flexible regimes.

In addition, an improved ant colony system with a new pheromone updating rule is proposed and used in the ODSD model, which considers the heuristic information of difference edges that comes from the comparison of two adjacent iteration-best solutions. The comparison experiment between the ACS and the ACSd shows that the ACSd has better performance, which proves that the difference edge strategy provides more exploration to avoid premature convergence and improve the solution quality.

Furthermore, the ACSd has been embedded with new heuristic information considering energy-efficient heuristic factor and speed heuristic factor in favor of a compromise between energy consumption and timing in solving the model. A case study also demonstrates that the proposed model, in terms of improving the flexibility of regime selection, reduces the energy consumption compared with the other methods.

In this paper, we focus on the feasibility of the model itself; therefore, only three traditional regimes are considered. In fact, the cruise regime should be included in our further model in our next work to make the ODSD model suitable for most main line railways. Besides, we will improve the ODSD model to make it better to meet more constraints for actual operation and consider the factors such as regenerative braking, passenger comfort, and passenger flow change. In terms of the algorithm, the strategy of difference edges can not only be integrated into the original ACS algorithm but also allowed to be integrated into other advanced algorithms.

Notations

Variables and Functions

E: Total energy consumption
I: The set of driving regimes, $I = \{r| -1, 0, 1\}$
i: Equal time subsegment index, $i = 1, 2, \ldots, N$, where N is the total subsegment number
ite_max: The maximum iteration number
L: Total trip length
L_q: Length of slope q
m: Number of ants
q: Slope index, $q = 1, 2, \ldots, Q$
$R_{i,u}$: An edge of a route, which represents the regime u in SE_i
r_i: Driving regime of the train, and $r_i \in \{-1, 0, 1\}$, where -1, 0, and 1 represent maximum braking, coasting, and maximum traction in SE_i, respectively
S_{Dif}: Set of difference edges
S_i: Running distance of SE_i
s: Iteration counter, $s = 1, 2, \ldots,$ ite_ max
SE_i: Equal time subsegment i
T: Total trip time
v_{ir}: Running speed at the end of SE_i with regime r
$\overline{v}_q$: The reference speed of slope q
$v_{q,\max}$: Speed limit of slope q
v_0, v_N: Starting speed and terminal speed
x_i: Position at the end of SE_i
x_0, x_N: Starting position and terminal position of the section
ΔS: The error between the total calculated distance and total trip length
$\Delta\tau_{\mathrm{iu}}^{\mathrm{best}}$: Pheromone increment on edge $R_{i,u}$ of the iteration best route
$\Delta\tau_{\mathrm{iu}}^{\mathrm{Nbest}}$: Extra pheromone added to edge $R_{i,u}$ in S_{Dif}
β: Weighted value of heuristic information to pheromone
δ: Allowable error between the total calculated distance and total trip length
η_{iu}: Heuristic information of $R_{i,u}$
η_1: Energy-efficient heuristic factor
η_2: Speed heuristic factor
ρ: Evaporation rate of the global pheromone
τ_{iu}: Pheromone accumulation on $R_{i,u}$

Abbreviations

ODSD: Optimal driving strategy decision-making
ACS: Ant colony system
ACSd: Ant colony system with the difference edges
TSP: Traveling salesman problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (grant no. 51478480) and the Hunan Provincial Natural Science Foundation of China (grant no. 12JJ3040). The authors would like to express their gratitude

to the supervisor Prof. Kun Miao for his great support and guidance in this project. They also thank the research team for their collaboration and help during gathering data for the research.

References

[1] National Bureau of Statistics, *China Energy Statistical Yearbook 2020*, China Statistics Press, Beijing, China, 2020.

[2] National Railway Administration, *Railway Statistical Bulletin 2020*, State Railway Administration, Beijing, China, 2021.

[3] K. Keskin and A. Karamancioglu, "Energy-efficient train operation using nature-inspired algorithms," *Journal of Advanced Transportation*, vol. 2017, Article ID 6173795, 12 pages, 2017.

[4] K. Ichikawa, "Application of optimization theory for bounded state variable problems to the operation of train," *Bulletin of JSME*, vol. 11, no. 47, pp. 857–865, 1968.

[5] A. Albrecht, P. Howlett, P. Pudney, X. Vu, and P. Zhou, "The key principles of optimal train control—Part 2: existence of an optimal strategy, the local energy minimization principle, uniqueness, computational techniques," *Transportation Research Part B: Methodological*, vol. 94, pp. 509–538, 2016.

[6] A. Albrecht, P. Howlett, P. Pudney, X. Vu, and P. Zhou, "The key principles of optimal train control—Part 1: formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points," *Transportation Research Part B: Methodological*, vol. 94, pp. 482–508, 2016.

[7] X. Jia, X. Zhou, J. Bao, G. Zhai, and R. Yan, "Fusion swarm-intelligence-based decision optimization for energy-efficient train-stopping schemes," *Applied Sciences*, vol. 13, no. 3, p. 1497, 2023.

[8] Y. Rao, P. Sun, Q. Wang, B. Bai, and X. Feng, "Optimal running time supplement for the energy-efficient train control considering the section running time constraint," *IET Intelligent Transport Systems*, vol. 16, no. 5, pp. 661–674, 2022.

[9] G. M. Scheepmaker and R. M. P. Goverde, "Multi-objective railway timetabling including energy-efficient train trajectory optimization," *European Journal of Transport and Infrastructure Research*, vol. 21, no. 4, pp. 1–42, 2021.

[10] Milroy and P. Ian, *Aspects of Automatic Train Control*, ian peter milroy, Adelaide, Australia, 1980.

[11] D. H. Lee, I. P. Milroy, and K. Tyler, *Application of Pontryagin's Maximum Principle to the Semi-automatic Control of Rail Vehicles*, National Conference Publication- Institution of Engineers, Barton, Australia, 1982.

[12] J. Cheng, Y. Davydova, P. Howlett, and P. Pudney, "Optimal driving strategies for a train journey with non-zero track gradient and speed limits," *IMA Journal of Management Mathematics*, vol. 10, no. 2, pp. 89–115, 1999.

[13] R. Liu and I. M. Golovitcher, "Energy-efficient operation of rail vehicles," *Transportation Research Part A: Policy and Practice*, vol. 37, no. 10, pp. 917–932, 2003.

[14] A. Albrecht, P. Howlett, P. Pudney, and X. Vu, "Optimal train control: analysis of a new local optimization principle," in *Proceedings of the 2011 American Control Conference*, San Francisco, CA, USA, June 2011.

[15] A. R. Albrecht, P. G. Howlett, P. J. Pudney, and X. Vu, "Energy-efficient train control: from local convexity to global optimization and uniqueness," *Automatica*, vol. 49, pp. 3072–3078, 2013.

[16] B. Mao, S. Chen, H. Liu, and T. K. Ho, "A simulation-based study for higher speed trains on busy railway mainlines," in *Proceedings of the International Conference on Applications of Advanced Technologies in Transportation Engineering*, pp. 305–312, ASCE- American Society of Civil Engineers, Washington, DC, USA, June 2002.

[17] X. Feng, "Optimization of target speeds of high-speed railway trains for traction energy saving and transport efficiency improvement," *Energy Policy*, vol. 39, no. 12, pp. 7658–7665, 2011.

[18] S. Lu, S. Hillmansen, T. K. Ho, and C. Roberts, "Single-train trajectory optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 743–750, 2013.

[19] H. Sun, J. Wu, H. Ma, X. Yang, and Z. Gao, "A Bi-objective timetable optimization model for urban rail transit based on the time-dependent passenger volume," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 604–615, 2019.

[20] N. Zhao, C. Roberts, S. Hillmansen, and G. Nicholson, "A multiple train trajectory optimization to minimize energy consumption and delay," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2363–2372, 2015.

[21] A. S Ghiduk and A. Alharbi, "Generating of test data by harmony search against genetic algorithms," *Intelligent Automation & Soft Computing*, vol. 36, no. 1, pp. 647–665, 2023.

[22] J. Eaton, S. Yang, and M. Gongora, "Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2980–2992, 2017.

[23] B. R. Ke, C. L. Lin, and C. W. Lai, "Optimization of train-speed trajectory and control for mass rapid transit systems," *Control Engineering Practice*, vol. 19, no. 7, pp. 675–687, 2011.

[24] X. Wang, Z. Guo, H. Zhang, C. Wang, and Y. Wang, "Snowmelt detection on the Antarctic ice sheet surface based on XPGR with improved ant colony algorithm," *International Journal of Remote Sensing*, vol. 44, no. 1, pp. 142–156, 2023.

[25] A. Rocha, A. Araujo, A. Carvalho, and J. Sepulveda, "A new approach for real time train energy efficiency optimization," *Energies*, vol. 11, no. 10, p. 2660, 2018.

[26] J. Pei, L. Xu, Y. Huang et al., "A two-step simulated annealing algorithm for spectral data feature extraction," *Sensors*, vol. 23, no. 2, p. 893, 2023.

[27] M. Domínguez, A. Fernández-Cardador, A. P. Cucala, T. Gonsalves, and A. Fernández, "Multi objective particle swarm optimization algorithm for the design of efficient ato speed profiles in metro lines," *Engineering Applications of Artificial Intelligence*, vol. 29, 2014.

[28] R. Geng, R. Ji, and S. Zi, "Research on task allocation of UAV cluster based on particle swarm quantization algorithm," *Mathematical Biosciences and Engineering*, vol. 20, no. 1, pp. 18–33, 2022.

[29] T. K. Ho, "Dynamic coast control of train movement with genetic algorithm," *International Journal of Systems Science*, vol. 35, no. 13-14, pp. 835–846, 2004.

[30] M. A. Sandidzadeh and M. R. Alai, "Optimal speed control of a multiple-mass train for minimum energy consumption using ant colony and genetic algorithms," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 231, no. 3, pp. 280–294, 2017.

[31] D. He, L. Zhang, S. Guo, Y. Chen, S. Shan, and H. Jian, "Energy-efficient train trajectory optimization based on improved differential evolution algorithm and multi-particle model," *Journal of Cleaner Production*, vol. 304, Article ID 127163, 2021.

[32] S. Zhan, P. Wang, S. C. Wong, and S. M. Lo, "Energy-efficient high-speed train rescheduling during a major disruption,"

Transportation Research Part E: Logistics and Transportation Review, vol. 157, Article ID 102492, 2022.

[33] D. He, G. Lu, and Y. Yang, "Research on optimization of train energy-saving based on improved chicken swarm optimization," *IEEE Access*, vol. 7, pp. 121675–121684, 2019.

[34] J. Yin, T. Tang, L. Yang, J. Xun, Y. Huang, and Z. Gao, "Research and development of automatic train operation for railway transportation systems: a survey," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 548–572, 2017.

[35] Y. Cao, Z. C. Wang, F. Liu, P. Li, and G. Xie, "Bio-inspired speed curve optimization and sliding mode tracking control for subway trains," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6331–6342, 2019.

[36] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[37] P. V. Matrenin, "Improvement of ant colony algorithm performance for the job-shop scheduling problem using evolutionary adaptation and software realization heuristics," *Algorithms*, vol. 16, no. 1, p. 15, 2022.

[38] W. Deng, J. Xu, and H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," *IEEE Access*, vol. 7, pp. 20281–20292, 2019.

[39] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[40] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1230–1242, 2013.

[41] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[42] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

[43] C. Zhu, J. Lu, and X. Li, "Review of studies on energy-efficient train operation in high-speed railways," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 18, no. 3, pp. 451–462, 2023.

[44] Y. Afek, R. Kecher, and M. Sulamy, "Optimal and resilient pheromone utilization in ant foraging," 2015, https://arxiv.org/abs/1507.00772.

[45] A. Acharya, D. Maiti, A. Konar, and R. Janarthanan, "A deterministic model for analyzing the dynamics of ant system algorithm and performance amelioration through a new pheromone deposition approach," 2008, https://arxiv.org/ftp/arxiv/papers/0811/0811.0080.pdf.

[46] P. B. Myszkowski, M. E. Skowroński, Ł. P. Olech, and K. Oślizło, "Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem," 2016, https://arxiv.org/abs/1603.08538.

[47] N. Ivkovic, R. Kudelic, and M. Golub, "Adjustable pheromone reinforcement strategies for problems with efficient heuristic information," *Algorithms*, vol. 16, no. 5, p. 251, 2023.

[48] N. Ivkovic, M. Malekovic, and M. Golub, "Extended trail reinforcement strategies for ant colony optimization," in *Swarm, Evolutionary, and Memetic Computing*, B. K. Panigrahi, P. N. Suganthan, S. Das, and S. C. Satapathy, Eds., pp. 662–669, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[49] B. Benjamin, I. Milroy, and P. Pudney, "Energy-efficient operation of long-haul trains," in *Proceedings of the Fourth International Heavy Haul Railway Conference 1989 Railways in Action Preprints of Papers*, Barton, Australia, July 1989.

[50] G. M. Scheepmaker, R. M. P. Goverde, and L. G. Kroon, "Review of energy-efficient train control and timetabling," *European Journal of Operational Research*, vol. 257, no. 2, pp. 355–376, 2017.

[51] X. Yang, X. Li, B. Ning, and T. Tang, "A survey on energy-efficient train operation for urban rail transit," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 2–13, 2016.

[52] J. Ning, Q. Zhang, C. Zhang, and B. Zhang, "A best-path-updatinginformation-guided ant colony optimization algorithm," *Information Sciences*, vol. 433, pp. 142–162, 2018.

[53] P. Chootinan and A. Chen, "Constraint handling in genetic algorithms using a gradient-based repair method," *Computers & Operations Research*, vol. 33, no. 8, pp. 2263–2281, 2006.

[54] W. Davis, *The Tractive Resistance of Electric Locomotives and Cars*, General Electric Review, Coimbatore, Tamil Nadu, 1926.

[55] N. Ivković, R. Kudelić, and M. Črepinšek, "Probability and certainty in the performance of evolutionary and swarm optimization algorithms," *Mathematics*, vol. 10, no. 22, p. 4364, 2022.

[56] N. Ivkovic, D. Jakobovic, and M. Golub, "Measuring performance of optimization algorithms in evolutionary computation," *International Journal of Machine Learning and Computing*, vol. 6, no. 3, pp. 167–171, 2016.

[57] Y. P. Fu, "Research on modeling and simulations of train tracking operation and saving energy optimization," Ph.D. dissertation, Beijing Jiaotong Univ, Beijing, China, 2009.