Single-Train Trajectory Optimization

Shaofeng Lu, *Member, IEEE*, Stuart Hillmansen, Tin Kin Ho, and Clive Roberts

***Abstract*—An energy-efficient train trajectory describing the motion of a single train can be used as an input to a driver guidance system or to an automatic train control system. The solution for the best trajectory is subject to certain operational, geographic, and physical constraints. There are two types of strategies commonly applied to obtain the energy-efficient trajectory. One is to allow the train to coast, thus using its available time margin to save energy. The other one is to control the speed dynamically while maintaining the required journey time. This paper proposes a distance-based train trajectory searching model, upon which three optimization algorithms are applied to search for the optimum train speed trajectory. Instead of searching for a detailed complicated control input for the train traction system, this model tries to obtain the speed level at each preset position along the journey. Three commonly adopted algorithms are extensively studied in a comparative style. It is found that the ant colony optimization (ACO) algorithm obtains better balance between stability and the quality of the results, in comparison with the genetic algorithm (GA). For offline applications, the additional computational effort required by dynamic programming (DP) is outweighed by the quality of the solution. It is recommended that multiple algorithms should be used to identify the optimum single-train trajectory and to improve the robustness of searched results.**

***Index Terms*—Ant colony optimization (ACO), dynamic programming (DP), energy saving strategy, rail traction systems, single-train trajectory.**

I. Introduction

DRIVER guidance systems [1] or automatic train operation [2] systems are able to take advantage of precomputed train speed trajectories. Train trajectory optimization has already been widely studied using various algorithms. Generally, train running trajectory optimization can be categorized into two types: coasting control and general control. Coasting control optimization searches for the optimum train trajectory by varying the coasting margin to use up the allowable time margin. A genetic algorithm (GA) has been applied in the search for the coasting points where the number of coasting points is predetermined [3]. The results demonstrate promising performance of coasting control for the tradeoff between the journey time and energy consumption. In [4], a GA was also applied to search for the coasting points. The number of coasting points has been dynamically allocated into the chromosomes, and this will enhance its practical application. In [5], some of the classic search methods, i.e., golden search methods, are studied in a simple single coasting point case supplementing the study of GAs. Artificial neural networks and GAs have been applied for the optimization of coasting points for trains [6]. Rather than searching for the coasting point, [7] targets the acceleration rate, the braking rate, and the remotoring speed.

The general control optimization derives the optimum train trajectory by applying a variety of sequential control inputs, i.e., acceleration, coasting, cruising, and deceleration. This means of optimization can be practically implemented in a straightforward manner because the control inputs are echoed by the practical train operations. Optimum control theory is among the widely applied techniques to obtain the optimum train trajectory. The objective is to operate a train and to minimize the energy consumption subject to time and other physical constraints. The solution of the problem is obtained through some linear approximation or some empirical extensions [8]. The Pontryagin maximum principle is the common method used to compute the solution in cases where the input signals are either continuous or discrete [9]. Because the methods using optimal control theories can be integrated with the fast response characteristics, they can be applied to develop online optimum control systems [8], [10]. Dynamic programming (DP) has been applied to search for the optimum trajectory with the minimum energy cost [11]–[13]. In [14], multipopulation GAs, together with the heuristic annealing selection, are applied to an urban railway vehicle. It is argued that a multiple population search improves the convergence rate and evolution stability.

However, it is not possible to obtain the analytical control input due to the nonlinear characteristics of the rail system. Some reasonable approximations should be taken to search for the optimum control signal and resultant trajectories. For example, in [8], the energy cost of the journey is assumed to rise linearly with the journey time, which may become unrealistic in practice. Some of the solutions in partial accelerating and braking control cannot be guaranteed to be optimal due to the singular characteristics of the train trajectory optimization [15], [16]. In addition, the optimal solution is not guaranteed, and convergence speed is uncertain in general in a numerical method [17], [18].

In an attempt to avoid the nonlinear complexity arising from the optimal control theory, this paper proposes a new graphic model based on which more general optimization algorithms can be applied and studied comparatively. Two heuristic

Manuscript received October 21, 2011; revised March 24, 2012, June 4, 2012, and November 2, 2012; accepted December 6, 2012. Date of publication January 14, 2013; date of current version May 29, 2013. This work was supported in part by the Advantage West Midlands and in part by the European Regional Development Funding through the Science City Energy Efficiency Project. The Associate Editor for this paper was B. Ning.

S. Lu was with the School of Electronic, Electrical and Computer Engineering, the University of Birmingham, Birmingham, B15 2TT, U.K. He is now with the Energy Research Institute, Nanyang Technological University, Singapore 639798 (e-mail: shaofeng.lu1@gmail.com).

S. Hillmansen and C. Roberts are with the School of Electronic, Electrical and Computer Engineering, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: s.hillmansen@bham.ac.uk; c.roberts.20@bham.ac.uk).

T. K. Ho is with the SMART Infrastructure Facility, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: markho@uow.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2012.2234118

algorithms and DP are applied to search for the train speed trajectory. The practical constraints are taken into account including the timetable, traction equipment characteristics, train operation speed limits, and gradients.

The study proposed in this paper focuses on the optimal speed trajectory of a single train with various scheduled journey times. The effects on the optimal trajectory imposed by other trains in the railway network are out of the scope of this paper. However, the searching algorithms are capable of accommodating situations where service disturbance is unexpectedly imposed as long as the initial and final train speed and time is known [19]. Broad readership can benefit from this paper in the application of driving guidance systems and other dynamic process optimization, such as network management, power distribution, shipping routes, etc.

The content of this paper is organized as follows. In Section II, the modeling procedure for the distance-based speed searching space of optimum running trajectory will be introduced. In Sections III–V, three varieties of algorithms are discussed based on the searching space model. In Section VI, the optimized train trajectory achieved from the three algorithms will be discussed comparatively. Finally, a conclusion will be drawn in Section VII.

II. Modeling Context

A. Vehicle Motion Modeling

The movement of a railway vehicle is determined by a set of physical constraints such as a journey profile, a speed limit, and other vehicle-related factors. The general equation of train motion, which is known as Lomonossoff's equation, can be written as follows:

$$M'\frac{d^2s}{dt^2} = F - \left(A + B\frac{ds}{dt} + C\frac{d^2s}{dt^2}\right) - Mg\sin(\alpha) \quad (1)$$

where F is the tractive effort or braking effort if applicable within the adhesion limit; A, B, and C are Davis constants; M' is the effective mass including rotary allowance; M is the tare mass; t is the dependent element time; s instant distance of the train; α is the slope angle.

A single-train motion simulator has been applied to calculate the energy consumption and the time cost of train movements. The energy consumption is calculated by the product between tractive effort distance. The calculation has considered the train characteristics, such as load and motor characteristics, and the route information, including the speed and gradient profiles. Further details about train energy and time calculation can be found, for instance, in [20]–[22].

The vehicle traction system prototype in this paper is based on the British Rail "Voyager" type. The speed and gradient profiles are shown in Fig. 1, whereas the information about the load and motor characteristics is shown in Fig. 2 and Table I.

B. Objective Function and Problem Formulation

This paper adopts distance-based modeling to simulate the train motion and traction power consumption. The motion of the train is calculated iteratively based on distance.

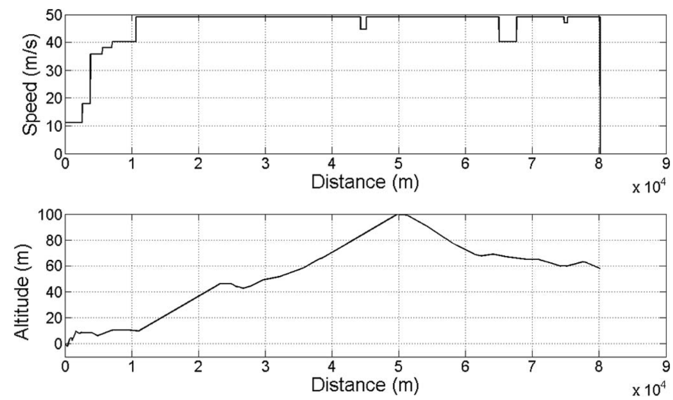


Fig. 1. Journey altitude and speed limit profile.

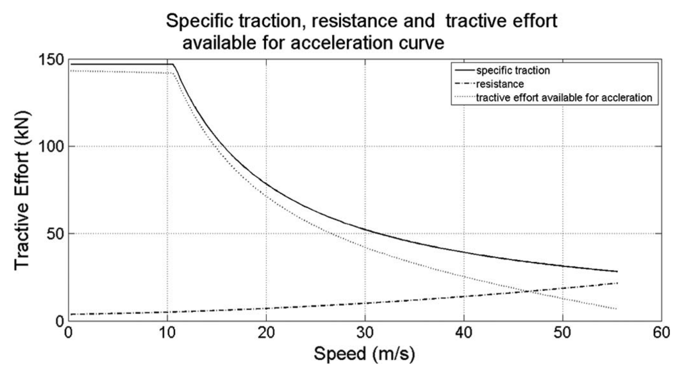


Fig. 2. Maximum tractive effort, resistance and acceleration curve of Voyager type vehicle.

The objective of the train trajectory optimization is to search for the speed for each position along the journey and to minimize the energy consumption subject to the punctuality requirement. The objective function to be minimized is defined as

$$J_{\text{tra}} = E(v_1, v_2, \ldots, v_n) + P \quad (2)$$

where E is the energy consumption for the proposed journey trajectory defined by a set of candidate speed $v_1, v_2, \ldots, v_n$ at the preset positions, T_{sched} is the scheduled journey time, and T_{srch} is the time cost for searched trajectory. P is the penalty cost related to the absolute difference ratio ϕ defined as $\phi = (|T_{\text{sched}} - T_{\text{srch}}|/T_{\text{sched}})$. The definition of P is listed in Table II.

The term "preset position" is used to describe the position at which the speed of vehicle needs to be determined, as shown in Fig. 3. The preset positions are classified into the following three types:

- positions whose distance values are the multiples of the proposed distance interval s_{int}, e.g., S_2 and S_3 preset positions in Fig. 3;
- positions at which the speed limits are changed, e.g., S_5 and S_7 preset positions in Fig. 3;
- positions for the beginning and the end of a journey, e.g., S_1 and S_{11} preset positions in Fig. 3.

TABLE I
KEY PARAMETERS FOR SINGLE TRAIN MOTION SIMULATOR

Tare mass (tonnes)	Maximum power (kW)	Maximum tractive effort (kN)	Davis coefficients A (kN)	B ($\frac{\text{kN}}{\text{m/s}}$)	C ($\frac{\text{kN}}{(\text{m/s})^2}$)
213.19	1568	146.8	3.73	0.0829	0.0043

TABLE II
DEFINITION OF PENALTY COST IN THREE ALGORITHMS

ϕ	GA and ACO	DP
$0 \leq \phi \leq 0.01$	0	0
$0.01 < \phi \leq 0.1$	E	∞
$0.1 < \phi \leq 0.2$	$30 \cdot \phi \cdot E$	∞
$0.2 < \phi$	$100 \cdot \phi \cdot E$	∞

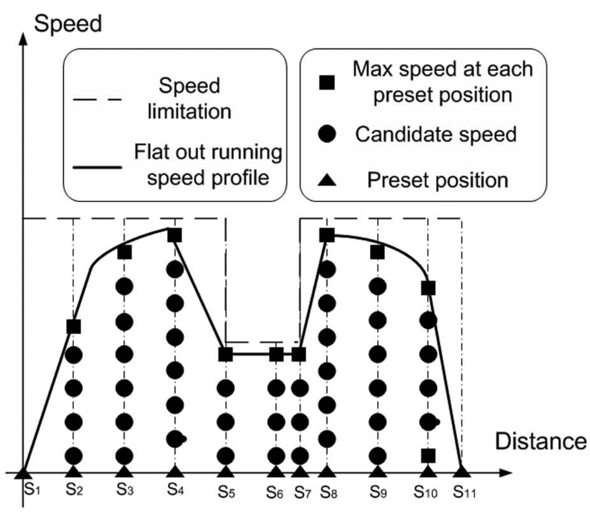


Fig. 3. Speed selection procedure in the distance based trajectory searching.

For each preset position, the possible speed should meet the following constraints.

- At each preset position, the maximum speed is determined by the "flat out" running of a train, during which the train runs as fast as possible without violating the speed limit.
- Between two preset positions, the calculation of the train trajectory is to use a minor distance step to calculate the actual energy consumption and time cost.

The optimization procedure is to search a set of candidate speed at each preset positions along the journey. Typical combinatorial optimization algorithms can therefore be applied. To generalize the optimization procedure, a construction graph covered in Section III is built to provide necessary information for different combinatorial optimization algorithms.

C. Construction of Graph

A complete weighted and directed graph $G = (N, A)$ is constructed with a set of N nodes, which are the train state, including the candidate speed and distance, and A being the set of arcs connection between the nodes. The energy consumption and time cost for each arc is calculated using a single-train simulator, as discussed in detail in [20] and [22].

Some remarks are made as follows.

- Let EC denote the sparse matrix to store the energy consumption for a train switch from one node to the other. $\text{EC}(i, j)$ are set as zero for an unfeasible switch or a braking switch between nodes i and j.
- Let TC denote the sparse matrix to store the time consumption when the train is switching between two nodes. Otherwise, zero will be stored.
- Let ECH denote the sparse matrix to store the heuristic energy consumption of every two nodes. $\text{ECH}(i, j)$ are set as zero for an unfeasible state switch between nodes i and j, and $|1/\text{EC}(i, j)|$ for connected nodes if otherwise.
- Let TCH denote the sparse matrix to store the heuristic time consumption between every two nodes. $\text{TCH}(i, j)$ are set as zero for an unfeasible state switch between nodes i and j, and $1/\text{TC}(i, j)$ for connected nodes if otherwise.
- Let LNK denote the linkage information sparse matrix. The linkage information $\text{LNK}(i, j)$ is used to indicate the feasibility and desirability of the switch between these two nodes.

III. ANT COLONY OPTIMIZATION

A. Introduction

Ant colony optimization (ACO) is inspired by the foraging behavior of the ant colony [23]. In ACO, a set of artificial ants communicates and cooperates indirectly by pheromone to find a solution to a discrete optimization problem. Each artificial ant, as an independent agent, is allocated with the computational resources by which it is able to leave the pheromone when necessary to communicate with the other ants. The ant with the good solution tends to leave more pheromone along their routes to direct the other ant. Using this "learning enhancement" style algorithm, the route with a better solution will finally attract more ants to follow and finally lead to a convergence of the optimization process. In [24], the max–min ant system, which is one type of the ACO algorithm, is applied to optimize the block layout for energy efficiency of mass rapid transit systems.

B. Solution Construction

The original pheromone trail imposed for every two connectable nodes is constant c_{lnk}, as shown in Algorithm 1.

At each construction step, ant "k" chooses the next speed at the next preset position based on a random proportional rule [23]. Assume that the ant is currently at the speed index i, and the possibility of speed index j being selected for the next preset position is defined as follows:

$$p_{i,j}^{k} = \frac{[\mathrm{LNK}(i,j)]^{\alpha}\,[\mathrm{ECH}(i,j)]^{\beta}\,[\mathrm{TCH}(i,j)]^{\gamma}}{\sum_{n\in\Omega_i^k}[\mathrm{LNK}(i,n)]^{\alpha}\,[\mathrm{ECH}(i,n)]^{\beta}\,[\mathrm{TCH}(i,n)]^{\gamma}} \quad (3)$$

where $\mathrm{LNK}(i,j)$, $\mathrm{ECH}(i,j)$, and $\mathrm{TCH}(i,j)$ are defined in Section II-C. α, β, and γ are the parameters to determine the relative influence of the pheromone trail and the heuristic information, and Ω_i^k are the feasible neighborhood of ant "k" being at node i. If the train is not running quickly enough, γ will be set at a higher value to attract ants to choose the less time cost switch. If the train is running too fast, β will be set at a higher value to attract the ant to choose a more energy-efficient node switch. α remains constant in this case. More details on determining the parameters can be found in [25].

Each artificial ant is able to decide which is the next indexed speed for the next preset position, and finally, a resultant journey can be constructed, showing the speed of the train at each preset positions. The quality of the solution will be evaluated using objective function (2), and the pheromone trail will then be updated based on each constructed solution's quality. One of the key functions in the update procedure is to reinforce the better solution through imposing more pheromone trail.

C. Pheromone Update and Termination Condition

The pheromone trail matrix is updated using the output of (3). A generalized update procedure is adopted for a group of artificial ants.

Use n_a to denote the number of ants in a group. Let n_p be the number of preset positions. Use SOL to denote the constructed solution matrix in which each row element is a trajectory solution. An element in a row is the index of each node at each preset position. The number of elements in each row is equal to n_p. Use EVAL to denote the 1-D matrix to store the evaluation function output for each row of constructed solutions. Let UPD denote the update vector to update the pheromone trail.

The update procedure can be divided into two parts. The first part is illustrated in the pseudocode shown in Algorithm 1.

Algorithm 1 ACO Part I: Obtain the update vector UPD for each constructed solution in SOL

Require: $\mathrm{eval}_{\min} \leftarrow \min(\mathrm{EVAL})$
  for $i = 1$ to n_a **do**
    $\overline{\mathrm{eval}} = \mathrm{EVAL}(i) - \mathrm{eval}_{\min}$
    $\mathrm{UPD}(i) = 2 \cdot c_{\mathrm{lnk}} \cdot exp(-\overline{\mathrm{eval}})$
  end for

Note that min is the function that is used to obtain the minimum element from its input vector. exp is the exponential function. The second part is illustrated in the pseudocode shown in Algorithm 2.

Algorithm 2 ACO part II: update the pheromone trail matrix LNK using UPD and SOL

$\mathrm{LNK}(r_i, c_i) \leftarrow (1 - c_e)\mathrm{LNK}(r_i, c_i)$
for $i = 1$ to n_a **do**
  for $j = 1$ to $n_p - 1$ **do**
    $r_i \leftarrow \mathrm{SOL}(i, j)$
    $c_i \leftarrow \mathrm{SOL}(i, j+1)$
    $\mathrm{LNK}(r_i, c_i) \leftarrow \mathrm{LNK}(r_i, c_i) + \mathrm{UPD}(i)$
  end for
end for

The best solution searched so far, i.e., $\mathrm{sol}_{\mathrm{bsf}}$, will be stored and updated by the new solution if a lower evaluation function output can be achieved.

The termination condition is set by two criteria. First, the number of groups of ants exceeds the maximum allowable number. Second, $\mathrm{sol}_{\mathrm{bsf}}$ keeps unchanged for a selected number of iterations.

IV. Genetic Algorithm

A. Introduction

The GA as a population-based optimization does not require gradient information on the objective function and only uses the output of the function to guide the search for a better solution. As mentioned in Section I, GA has been reported as the successful candidate algorithm in various train running trajectory searching applications, and the simulation results shows its robustness in this area [3], [4], [7], [14]. Here, the GA will be used to search for the characterized speed at each preset position using the modeled strings i.e., genotypes. Each string is modeled as a characterized signal for current speed jump.

B. Genotype Generation

To apply the GA, two important steps should be implemented:

- generation of the population of strings (genotypes);
- creation of a fitness function to distinguish the better strings.

Note that, for each candidate speed at each preset position, the speeds in the neighborhood have a range. Various characterized operations can be identified through the speed switch.

At each section between two adjacent preset positions, a control index number is allocated. Assume that the speed at the former position is v_i and that v_j is the speed at the latter one. It is assumed that $v_j \in [v_{\min}, v_{\max}]$, where $v_{\min}$ and $v_{\max}$ are the minimum and maximum possible speed levels of v_j. i and j are the unique index number for both speeds. Let v_{ee} denote such speed that switches from v_i in a most energy-efficient operation with $\mathrm{ECH}(i,j) \neq \infty$, and let v_c denote such v_j for a coasting operation with $\mathrm{ECH}(i,j) = \infty$.

For the control index number of "0," the most energy-efficient state switch will be selected. For the control index of "1–7," six speeds in the range of $[V_{\min}, V_{\max}]$ will be selected

TABLE III
NEXT SPEED SELECTION BASED ON THE CHARACTERISED CONTROL INDEX NUMBER

Control index i_c	Next speed selected
0	v_{ee} or v_c whichever exists
$1 \sim 6$	$V_{min} + (V_{max} - V_{min}) \cdot \frac{i_c - 1}{5}$
7	V_{cur} if feasible

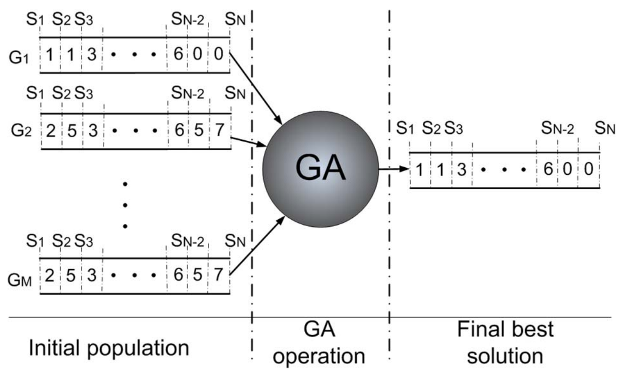


Fig. 4. Schematic GA optimization of train running trajectory. Given a string of control index, a trajectory solution is derived and it can be evaluated using the objective function as a fitness function for GA optimisation.

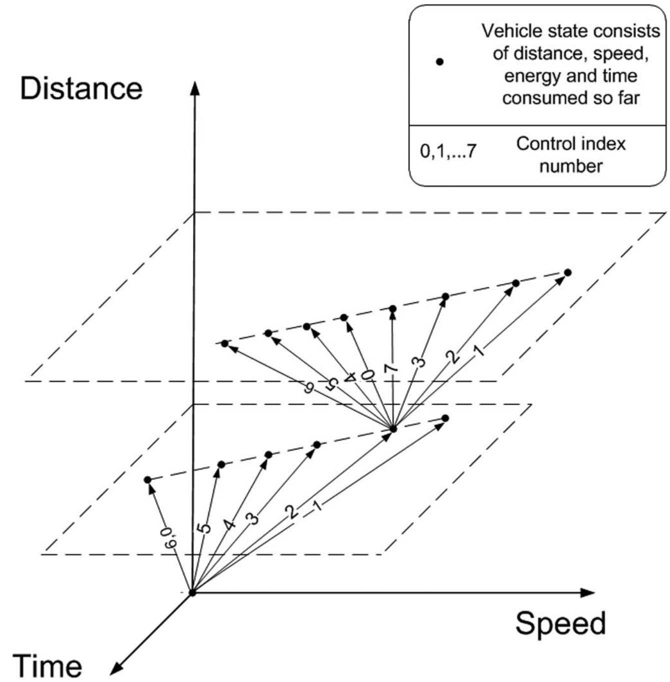


Fig. 5. States generation procedure in the DP algorithm.

using the methods shown in Table III. The control index number of "7" is the cruising operation of the vehicle if the speed is allowed to be kept in the next preset position.

Assume that there are M strings in the initial population, and N preset positions along the journey. The schematic GA optimization procedure is shown in Fig. 4.

V. DYNAMIC PROGRAMMING

A. Introduction

DP [26] is a powerful tool to solve a problem that can be divided into various substages. In our case, the trajectory searching can naturally be divided into subintervals of distance.

B. Optimization Process

Let us make the following definition first.

Vehicle state ϕ consists of four basic physical elements: vehicle distance s, vehicle speed v, used journey time t, and used energy consumption since the vehicle sets off from the initial state where $s = 0$ and $v = 0$.

Vehicle state can be expressed in an array form, i.e.,

$$\phi = [s, v, t, e]. \tag{4}$$

Let ϕ_o denote the initial vehicle state, and obviously, the following equation should hold:

$$\phi_o = [s_o, v_o, t_o, e_o] = [0, 0, 0, 0]. \tag{5}$$

According to the hypothesis, each vehicle state must have one of the preset positions as its instant distance.

DP is proceeded forward iteratively, as shown in Fig. 5. All the vehicle states are developed from the initial one, which is the original point in this graph. For the first step, the states will be created from the initial state using the index control signal mentioned in Section IV. The index control signal for cruising operation is not available for the first step. One of the examples for the second step is also presented, which should be applied for all the other created states from the initial state.

After the initial state has been created, two operations should be performed iteratively.

State Generation: Each of the states should be used to generate the next state, unless the preset position in the current state is equal to the final preset position. Take the initial state as an example. According to the philosophy of the characterized control index, assume that the current control index is u_1; thus, the following can be derived:

$$s_o \xrightarrow{u_1} s_1 \tag{6}$$

$$v_o \xrightarrow{u_1} v_1 \tag{7}$$

where s_1 will be the preset position right after s_o, whereas v_1 is determined using the method shown in Table III. The time and energy cost due to the distance and the speed switch can be found in the sparse matrices TC and EC. Assuming that the time cost is t_c and that the energy cost is e_c, we have

$$t_1 = t_0 + t_c \tag{8}$$

$$e_1 = e_0 + e_c. \tag{9}$$

New state s_1 is thereafter produced based on its parent state s_o. Each of the newly generated states is able to remember its parent state using the indexing method.

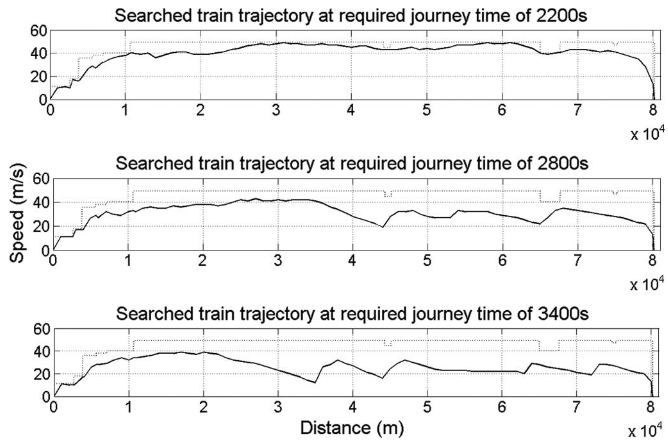


Fig. 6. Optimized journey trajectories using ACO under different journey time conditions.

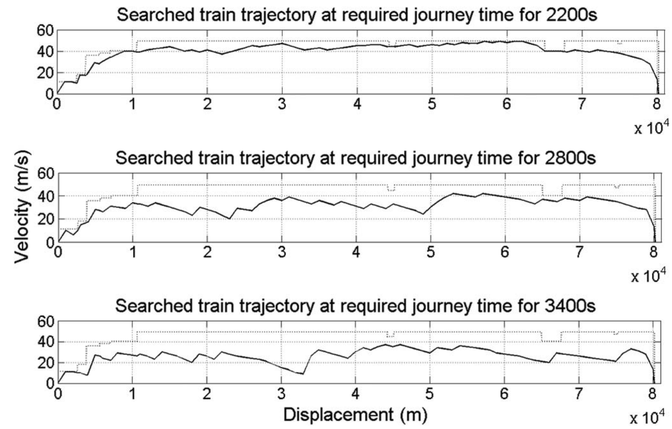


Fig. 7. Optimized journey trajectories using GA under different journey time conditions.

Elimination of Duplicate States: It is important that each of the states is accompanied with minimum energy caused so far since the state s_o. Duplicated state elimination occurs any time that there are two states with identical d, t, and v. The state with more energy cost will be therefore eliminated.

To reduce the actual number of generated states in the searching space, further action is taken to confine the actual searching space. The vehicle state that is outside of the admissible area will be ruled out from the searching. A simple heuristic is adopted: The instantaneous position of the train should not be significantly different from a position defined by the average speed [11]. Accordingly, we define the upper bound and the lower bound for the journey time cost at various journey distances.

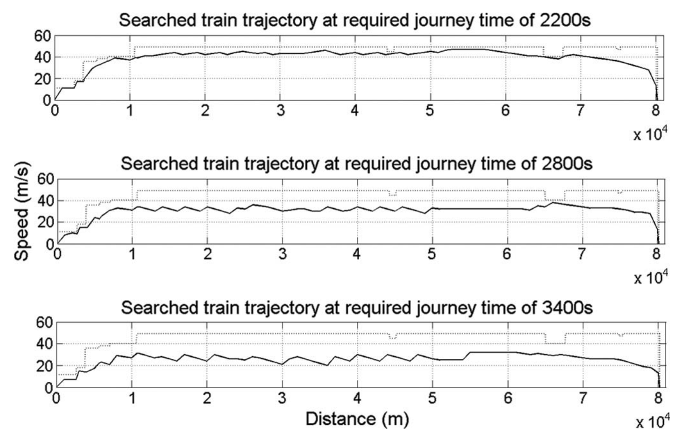


Fig. 8. Optimized journey trajectories using DP under different journey time conditions.

C. Summary

DP has been applied to search for the optimum journey trajectories in terms of augmented vehicle states routes. By dividing the searching procedure into different subintervals, DP is able to obtain the minimum energy cost for vehicle switching from its original state to the current state by eliminating the same states with more energy cost. An admissible area for the DP search has been adopted to reduce the total searching states. Any state that stands outside of the admissible area will be ruled out of the searching procedure. The concept of admissible area relies on the heuristic that, for a feasible solution of train trajectory, the instant position of a train cannot vary too much from the position defined by the average speed.

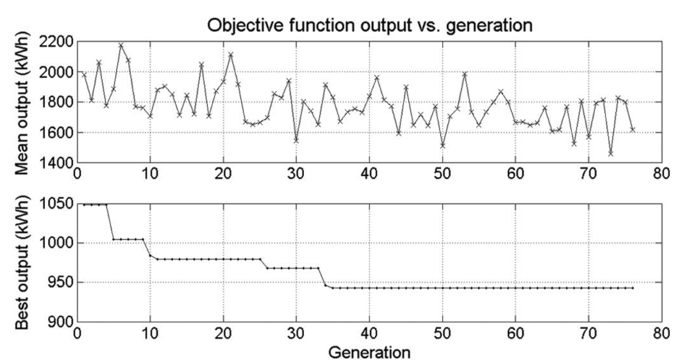


Fig. 9. Best and mean objective function output for journey time of 2800 s at each generation for ACO.

VI. Results and Discussion

The key simulation results are shown here. First, trajectories for various scheduled journey times, i.e., 2200, 2800, and 3400 s for ACO, GA, and DP, are presented in Figs. 6–8.

For journeys with considerable journey time margin, both the ACO algorithm and the GA fail to find a smooth trajectory. There is no extended cruising phase, and often, there is a considerable difference between the maximum and minimum speeds in the central part of the journey. These are clearly not good solutions. The DP, on the other hand, performed better with a more constant cruising speed below the line speed limit.

Figs. 9 and 10 show how the objective function output evolves with the generation for the journey time of 2800 s.

The journey time cost versus energy cost curves for different scheduled journey times are compared among the three algorithms. The journey time cost ranges from 2100 to 3500 s with an interval of 100 s. These curves are shown in Fig. 11. Note that each mark in the figure shows a combination of the journey

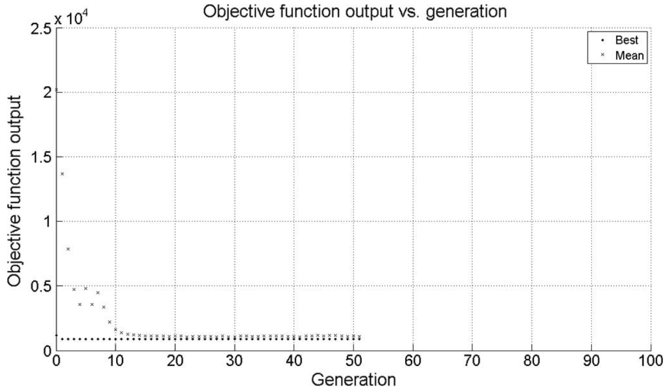


Fig. 10. Best and mean objective function output for journey time of 2800 s at each generation for GA.

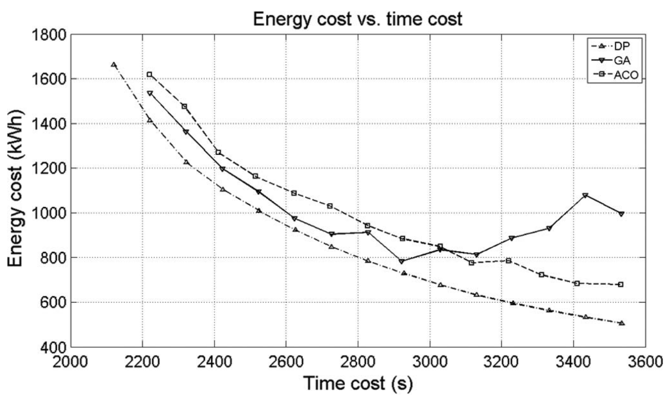


Fig. 11. Journey energy cost versus time cost curves using different algorithms.

time cost and energy cost for a simulation. The shape of the mark distinguishes the type of algorithm.

When the journey time constraint is small, all three algorithms perform well; however, it was only possible to reach a solution for 2100 s using the DP algorithm, because the others both failed to converge. At journey times greater than 2800 s, the performance of the GA significantly decreased. More heuristic information is used in the case of the ACO, and the algorithms' performance remains more stable than that of GA. It is demonstrated that an optimum solution is not guaranteed for heuristic algorithms, and the performance of heuristic algorithms can be significantly affected by searching space.

DP on the other hand is able to obtain the best solution among the three algorithms, but it requires significantly more computational resources. Since any combination of current used journey time, current used energy, and current distance implies a unique state in the searching space, the computational complexity becomes enormous. Such an algorithm demonstrates its robust searching capability, even at lower journey times, e.g., 2100 s. However, the algorithms based on the random Monte Carlo-style selection have the possibility of never finding a suitable solution.

Table IV shows a comparison of the characteristics between the three algorithms.

TABLE IV
CHARACTERISTICS COMPARISON BETWEEN THREE ALGORITHMS APPLIED ON THE JOURNEY WITH SCHEDULED TIME OF 2800 s FOR 15 RUNS

Algorithms	Mean value	Deviation (%)	Aver.comp.time (Unit)
ACO	946.6	16.6	1
GA	885	51.6	2.81
DP	784.6	0	4.4

VII. Conclusions and Summary

A. Overview

In this paper, methods for single-train trajectory optimization have been discussed. Choosing the sequence of control operations is a problem that requires a nontrivial solution. By approximating train running trajectory over a relatively short distance, the search for the sequence can be turned into the procedure of determining the speed at different preset positions. A sparse storage model is proposed. Two heuristic algorithms including the ACO algorithm and the GA are applied based on this model. DP is also used to search for the target speed, and the simulation results are demonstrated and discussed.

B. Conclusions

- The solution to the optimum train trajectory cannot be solved analytically, and numerical methods must be used.
- The performance of the three methods has been contrasted and compared. It was found that DP performed better than both GA and ACO. Under certain circumstances, the GA performed quite poorly and failed to converge onto a good solution (particularly for large journey times). It may be possible to tune the search algorithm, but without comparative results from alternative methods, it would be impossible to determine the existence of better solutions. ACO depended on strong heuristic information and performed adequately for most of the journey time allowances. It also arrived at a solution significantly quicker than the other methods.
- For those cases where the solution space becomes small, both the GA and ACO failed to converge on a solution.
- In general, it is recommended that more than one method should be used to identify the optimum trajectories because it is often possible to converge on a solution that is plausible, yet nowhere near optimal.

References

[1] I. Mitchell, "The sustainable railway-use of advisory systems for energy savings," *IRSE News*, no. 151, pp. 2–8, Dec. 2009, Tech. Paper.
[2] S.-H. Han, S.-G. Lee, S.-G. Kim, and W.-D. Lee, "Design of optimal control for automatic train operation system in Emu," in *Proc. IEEE ICCAS*, 2001, pp. 394–397.
[3] C. Chang and S. Sim, "Optimising train movements through coast control using genetic algorithms," *Proc. Inst. Elect. Eng.—Elect. Power Appl.*, vol. 144, no. 1, pp. 65–73, Jan. 1997.
[4] K. K. Wong and T. K. Ho, "Dynamic coast control of train movement with genetic algorithm," *Int. J. Syst. Sci.*, vol. 35, no. 13/14, pp. 835–846, Oct.–Nov. 2004.

[5] K. Wong and T. Ho, "Coast control for mass rapid transit railways with searching methods," *Proc. Inst. Elect. Eng.—Elect. Power Appl.*, vol. 151, no. 3, pp. 365–376, May 2004.

[6] S. Acikbas and M. Soylemez, "Coasting point optimisation for mass rail transit lines using artificial neural networks and genetic algorithms," *IET Elect. Power Appl.*, vol. 2, no. 3, pp. 172–182, May 2008.

[7] Y. V. Bocharnikov, A. M. Tobias, C. Roberts, S. Hillmansen, and C. J. Goodman, "Optimal driving strategy for traction energy saving on DC suburban railways," *IET Elect. Power Appl.*, vol. 1, no. 5, pp. 675–682, Sep. 2007.

[8] R. Liu and L. M. Golovitcher, "Energy-efficient operation of rail vehicles," *Transp. Res. A, Policy Practice*, vol. 37, no. 10, pp. 917–932, Dec. 2003.

[9] P. Howlett, "The optimal control of a train," *Ann. Oper. Res.*, vol. 98, no. 1–4, pp. 65–87, Dec. 2000.

[10] E. Khmelnitsky, "On an optimal control problem of train operation," *IEEE Trans. Autom. Control*, vol. 45, no. 7, pp. 1257–1266, Jul. 2000.

[11] H. Ko, T. Koseki, and M. Miyatake, "Application of dynamic programming to the optimization of the running profile of a train," in *Computers in Railway Six*, J. Allan, C. Brebbia, R. Hill, G. Sciutto, and S. Sone, Eds. Ashurst., U.K.: WIT Press, 2004, pp. 103–112.

[12] S. Effati and H. Roohparvar, "The minimization of the fuel costs in the train transportation," *Appl. Math. Comput.*, vol. 175, no. 2, pp. 1415–1431, Apr. 2006.

[13] R. Franke, P. Terwiesch, and M. Meyer, "An algorithm for the optimal control of the driving of trains," in *Proc. IEEE 39th Conf. Decision Control*, 2000, vol. 3, pp. 2123–2128.

[14] W. Liu, Q. Li, and B. Tang, "Energy saving train control for urban railway train with multi-population genetic algorithm," in *Proc. Int. Forum Inf. Technol. Appl.*, 2009, pp. 58–63.

[15] C. Johnson and J. Gibson, "Singular solutions in problems of optimal control," *IEEE Trans. Autom. Control*, vol. AC-8, no. 1, pp. 4–15, Jan. 1963.

[16] M. Athans and P. L. Falb, *Optimal control: An Introductions to the Theory and Its Applications*. New York: McGraw-Hill, 1966.

[17] M. Miyatake and H. Ko, "Optimization of train speed profile for minimum energy consumption," *IEEJ Trans. Elect. Electron. Eng.*, vol. 5, no. 3, pp. 263–269, May 2010.

[18] Y. Wang, B. Ning, F. Cao, B. De Schutter, and T. van den Boom, "A survey on optimal trajectory planning for train operations," in *Proc. IEEE Int. Conf. SOLI*, Jul. 2011, pp. 589–594.

[19] I. A. Hansen and J. Pachl, Eds., *Railway Timetable & Traffic : Analysis, Modelling, Simulation*. Hamburg, Germany: Eurailpress, 2008.

[20] S. Lu, S. Hillmansen, and C. Roberts, "A power-management strategy for multiple-unit railroad vehicles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 406–420, Feb. 2011.

[21] A. Mirabadi and M. Najafi, "Energy management system in hybrid trains using fuzzy control: A comparative study," *Proc. Inst. Mech. Eng. F, J. Rail Rapid Transit*, vol. 225, no. F3, pp. 267–276, May 2011.

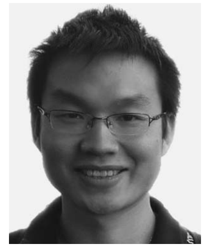
[22] S. Hillmansen and C. Roberts, "Energy storage devices in hybrid railway vehicles: A kinematic analysis," *Proc. Inst. Mech. Eng. F, J. Rail Rapid Transit*, vol. 221, no. 1, pp. 135–143, Jan. 2007.

[23] M. Dorigo and S. Thomas, *Ant Colony Optimisation*. Cambridge, MA: MIT Press, 2004.

[24] B.-R. Ke, M.-C. Chen, and C.-L. Lin, "Block-layout design using max-min ant system for saving energy on mass rapid transit systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 226–235, Jun. 2009.

[25] S. Lu, "Optimising power management strategies for railway traction systems," Ph.D. dissertation, Sch. Electron., Elect. Comput. Eng., Univ. Birmingham, West Midlands, U.K., 2011.

[26] R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957, ser. Rand Corp. Res. Study.



Shaofeng Lu (M'12) received the B.Eng. and Ph.D. degrees from the University of Birmingham, Birmingham, U.K., in 2007 and 2011, respectively, and the B.Eng. degree from Huazhong University of Science and Technology, Wuhan, China, all in electrical and electronic engineering.

He is currently a Postdoctoral Researcher with the Birmingham Centre for Railway Research and Education, School of Electronic, Electrical, and Computer Engineering, University of Birmingham. His main research interests include railway vehicle traction system modeling, power management strategies, railway electrical networks, and optimization algorithms.

Stuart Hillmansen received the Ph.D. degree from Imperial College London, London, U.K.

He is currently a Lecturer of electrical energy systems with the School of Electronic, Electrical, and Computer Engineering, University of Birmingham, Birmingham, U.K. He is also a member with the Birmingham Centre for Railway Research and Education, School of Electronic, Electrical, and Computer Engineering, University of Birmingham. He leads the Railway Traction Research Group, whose portfolio of activities is supported by the railway industry and government. He is the author of a number of papers on railway energy consumption and has presented work at a number of international conferences. His main research interests include hybrid traction systems for use in railway vehicles and modeling and measurement of energy consumption for railway systems (both alternating current and direct current).

Dr. Hillmansen is a member of the Editorial Board of the *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*.

Tin Kin Ho received the B.Eng. and Ph.D. degrees in electronic and electrical engineering from the University of Birmingham, Birmingham, U.K., in 1988 and 1994, respectively.

He is currently an Associate Professor with the SMART Infrastructure Facility, University of Wollongong, Wollongong, Australia. He is also an Honorary Professor with Beijing Jiaotong University, Beijing, China. His research interests include railway signaling and operation simulation, asset management and condition monitoring, rail open markets, and intelligent scheduling.

Clive Roberts is currently a Professor of railway systems and a Director for research with the Centre for Railway Research and Education, School of Electronic, Electrical, and Computer Engineering, University of Birmingham, Birmingham, U.K. His Ph.D. research focused on condition monitoring of railway infrastructure. Over the last 14 years, he has developed a portfolio of research in the fields of railway systems engineering, system modeling and simulation, network capacity research, railway fault detection and diagnosis, and data collection and decision support applied to railway traction, signaling, mechanical interactions, and capacity. He currently leads a team of 11 research fellows and 16 Ph.D. students.