Single Train Trajectory Optimisation

Shaofeng Lu, Stuart Hillmansen, Mark 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 required journey time. This paper proposes a distance based train trajectory searching model, upon which three optimisation 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 Algorithm (ACO) obtains better balance between stability and the quality of the results, in comparison to the other algorithms, Dynamic Programming (DP) and Genetic Algorithm (GA). For off-line applications, the additional computational effort required by dynamic programming is outweighed by the quality of the solution.**

***Index Terms*—energy saving strategy, single train trajectory, dynamic programming, ant colony optimisation, rail traction systems**

I. Introduction

Driver guidance systems [1] or Automatic Train Operation (ATO) [2] systems are able to take advantage of pre-computed train speed trajectories. Train trajectory optimisation has already been widely studied using various algorithms. Generally, the train running trajectory optimisation can be categorized into two types: coasting control and general control. The coasting control optimisation 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 point is predetermined [3]. The results demonstrate promising performance of coasting control for the tradeoff between the journey time and energy consumption. In work reported 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 work reported 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 GA. Artificial Neural Networks (ANN) and GA have been applied for the optimisation of coasting points for trains [6]. Rather than search for the coasting point, the work demonstrated in [7] targets on the acceleration rate, the braking rate and the re-motoring speed.

The general control optimisation derives the optimum train trajectory by applying a variety of sequential control inputs, i.e. acceleration, coasting, cruising and deceleration. This mean of optimisation 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 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]. Pontryagin Maximum Principle (PMP) is the common method used to compute the solution in cases where the input signal 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 an 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 work presented in [14], multi-population genetic algorithms together with the heuristic annealing selection is applied to an urban railway vehicle. It is argued that a multiple

Manuscript received Oct. 21th, 2011.

This research used equipments funded by AWM and ERDF through the Science City Energy Efficiency project. E-mail: Dr. Stuart Hillmansen (s.hillmansen@bham.ac.uk).

population search improves the convergence rate and evolution stability.

However, it is not possible to obtain the analytical control input due to the non-linear characteristic of the rail system. Some reasonable approximations should be taken in order to search for the optimum control signal and resultant trajectories. For example, in the work presented in [8], the energy cost of the journey is assumed to be rise linearly with the journey time which may become unpractical in practice. Some of solutions in partial accelerating and braking control cannot be guaranteed to be optimal due to the singular characteristic of the train trajectory optimisation [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 non-linear complexity arising from the optimal control theory, this paper proposes a new graphic model based on which more general optimisation algorithms can be applied and studied comparatively. Two heuristic algorithms and dynamic programming are applied to search for the train speed trajectory. The practical constraints are taken into account including the timetable, traction equipments characteristics, train operation speed limits, and gradients.

The study proposed in this paper is focused on the optimal speed trajectory of a single train with various scheduled journey time. The effects on the optimal trajectory imposed by other trains in the railway network are out of the scope of this study. 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 system and other dynamic process optimisation such as network managements, power distribution, shipping routes etc.

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

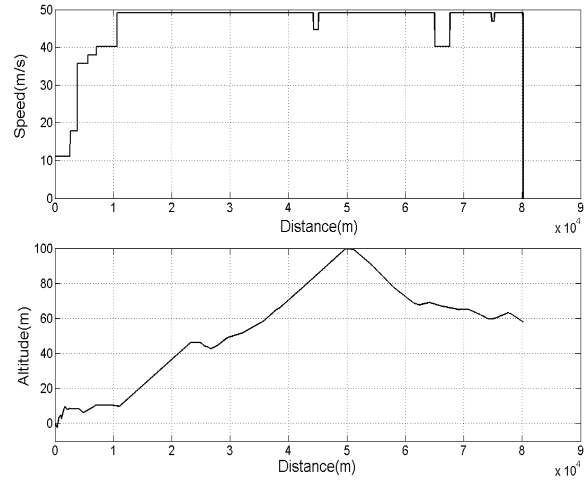


Fig. 1. Journey altitude and speed limit profile.

II. Modeling context

A. Vehicle motion modeling

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

$$M'\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} = F-(A+B\frac{\mathrm{d}s}{\mathrm{d}t}+C\frac{\mathrm{d}^2 s}{\mathrm{d}t^2})-Mg\sin(\alpha) \quad (1)$$

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 dependant element time;
- s is the instant distance of train;
- α is the slope angle.

A single train motion simulator has been applied to calculate the energy consumption and time cost of train movements. The energy consumption is calculated by tractive effort times distance. The calculation has considered the train characteristic, such as load and motor characteristic, route information including speed profile and gradient profile. Further details of train energy and time calculation can be found in various references [20]–[22].

The vehicle traction system prototype in this study is based on the British Rail "Voyager" type. The information about the load and motor characteristic is shown in Fig. 2 and Table I while the

TABLE I
KEY PARAMETERS FOR SINGLE TRAIN MOTION SIMULATOR

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

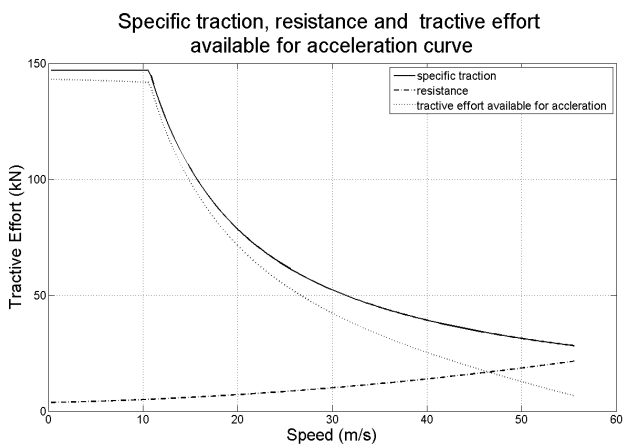


Fig. 2. Maximum tractive effort, resistance and acceleration curve of Voyager type vehicle

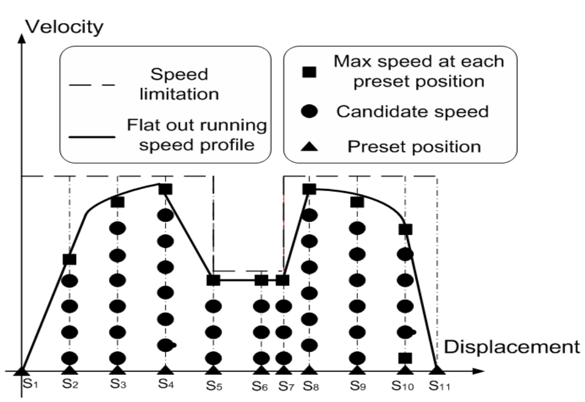


Fig. 3. Speed selection procedure in the distance based trajectory searching

speed and gradient profiles are shown in Fig. 1.

B. Objective function and problem formulation

This paper adopts the distance based modeling to simulate the train motion and traction power consumption. The motion of train is calculated iteratively based on distance. The objective of the train trajectory optimisation is to search for the speed for each position along the journey and minimise the energy consumption subject to the punctuality requirement. The objective function to be minimized is defined in 2.

$$J_{tra} = E(v_1, v_2, \cdots, v_n) + P \qquad (2)$$

where E is the energy consumption for the proposed journey trajectory defined by a set of candidate speed $v_1, v_2, \cdots, v_n$ 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 $\phi = \frac{|T_{sched} - T_{srch}|}{T_{sched}}$. The definition of P is listed in Table II.

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	∞
$0.1 < \phi \leq 0.2$	$30 \cdot \phi \cdot E$	∞
$0.2 < \phi$	$100 \cdot \phi \cdot E$	∞

The term "preset position" is used to describe the position at which the speed of vehicle need 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 journey, e.g. S_1 and S_{11} preset positions in Fig. 3.

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 optimisation procedure is to search a set of candidate speed at each preset positions along the journey. Typical combinatorial optimisation algorithms can therefore be applied. In order to generalise the optimisation procedure, a construction graph covered in next section is built to provide necessary information for different combinatorial optimisation algorithms.

C. Construction of graph

A complete weighted and directed graph $G = (N, A)$ is constructed with N being the 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 details in [20], [22].

Some remarks are made as follows:

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

III. Ant colony optimisation

A. Introduction

ACO is inspired by the foraging behavior of the ant colony [23]. In ACO, a set of artificial ants communicate and cooperate indirectly by pheromone to find a solution to a discrete optimisation 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 better solution will finally attract more ants to follow and finally lead to a convergence of the optimisation process. In [24], Max-Min ant system, one type of ACO algorithm, is applied to optimise the block layout for energy efficiency of mass rapid transit system.

B. Solution construction

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

At each construction step, ant "k" choose the next speed at 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 next preset position is defined as follows:

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

where $LNK(i, j)$, $ECH(i, j)$ and $TCH(i, j)$ are defined in II-C. α, β, 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 running not quickly enough γ will be set higher value to attract ants to choose less time cost switch. If the train is running too fast, β will be set higher value to attract ant to choose 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 next preset position and finally a resultant journey can be constructed showing the speed of train at each preset positions. The quality of the solution will be evaluated using the objective function (2) and the pheromone trail will then be updated based 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 generalised 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 being the number of preset positions. Use SOL to denote the constructed solution matrix in which each row element is a trajectory solution. A element in a row is the index of each node at each preset positions. The number of elements in each row equals to n_p. Use $EVAL$ to denote the one dimension matrix to store the evaluation function output for each row of constructed solutions. Let UPD to 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 pseudo-code shown in Alg. 1.

Algorithm 1 ACO part I: obtain the update vector UPD for each constructed solution in SOL

```
Require: eval_min ← min(EVAL)
  for i = 1 to n_a do
    eval‾ = EVAL(i) − eval_min
    UPD(i) = 2 · c_lnk · exp(−eval‾)
  end for
```

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

The best solution searched so far sol_{bsf} will be stored and updated by the new solution if lower evaluation function output can be achieved.

Termination condition is set by two criterions. Firstly, the number of groups of ants exceeds the maximum allowable number. Secondly, the sol_{bsf} keeps unchanged for a selected number of iterations.

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

```
LNK(r_i, c_i) ← (1 − c_e)LNK(r_i, c_i)
for i = 1 to n_a do
  for j = 1 to n_p − 1 do
    r_i ← SOL(i, j)
    c_i ← SOL(i, j + 1)
    LNK(r_i, c_i) ← LNK(r_i, c_i) + UPD(i)
  end for
end for
```

IV. Genetic algorithm

A. Introduction

Genetic Algorithm (GA) as a population based optimisation does not require gradient information of the objective function and only use the output of the function to guide the search for better solution. As mentioned in the introduction section I, GA has been reported as the successful candidate algorithm in various train running trajectory searching application and the simulation results shows its robustness in this area [3], [4], [7], [14]. In this section, the GA will be used to search for the characterised speed at each preset position using the modeled strings i.e. genotypes. Each strings is modeled as a characterised signal for current speed jump.

B. Genotype generation

In order to apply the GA, two important steps should be implemented.

- Generation of the population of strings (genotypes).
- Creation of fitness function to distinguish the better strings.

Notice that for each candidate speed at each preset position, the speeds in the neighborhood have a range. Various characterised operation 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 level

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

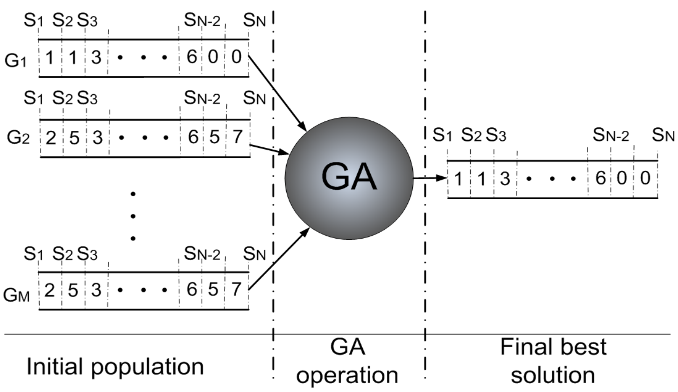


Fig. 4. Schematic Genetic Algorithm optimisation of train running trajectory. Given a string of control index, a trajectory solution is derived and it can be evaluated using the objective function as a fitness function for GA optimisation.

of v_j. i, 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 $ECH(i,j) \neq \infty$ and let v_c denote such v_j for a coasting operation with $ECH(i,j) = \infty$.

For the control index number of "0", the most energy-efficient states switch will be selected. For the control index of "1" to "7" , six speeds in the range of $[V_{min}, V_{max}]$ will be selected using the methods shown in Table III. The control index number of "7" is the cruising operation of vehicle if the speed is allowed to be kept in the next preset position.

Assume that there is "M" strings in the initial population, "N" preset positions along the journey, the schematic GA optimisation procedure is shown in Fig. 4.

V. Dynamic Programming

A. Introduction

Dynamic programming [26] is a powerful tool to solve a problem which can be divided into various sub-stages. In our case, the trajectory searching can naturally be divided into sub-intervals of distance.

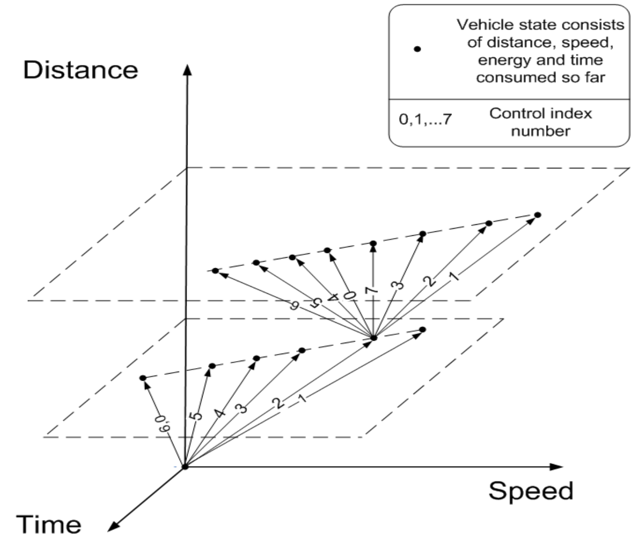


Fig. 5. States generation procedure in the Dynamic Programming algorithm

B. Optimisation 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

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

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.

The Dynamic Programming is proceeded forward iteratively 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 IV. Index control signal for cruising operation is not available for 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, 2 operations should be performed iteratively.

State generation Each of the states should be used to generate 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 the philosophy of the characterised control index, assume the current control index is u_1, 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, while v_1 is determined using the method shown in III. The time and energy cost due to the distance and speed switch can be found in the sparse matrix TC and EC. Assume the time cost is t_c and energy cost is e_c.

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

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

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.

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 there is 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. For the vehicle state which 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 different significantly from a position defined by the average speed [11]. Accordingly, we define the upper bound and lower bound for the journey time cost at various journey distance.

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 sub-intervals, DP is able to obtain the minimum energy cost for vehicle switching from its original state to 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 states which stand 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 train can not vary too much from the position defined by the average speed.

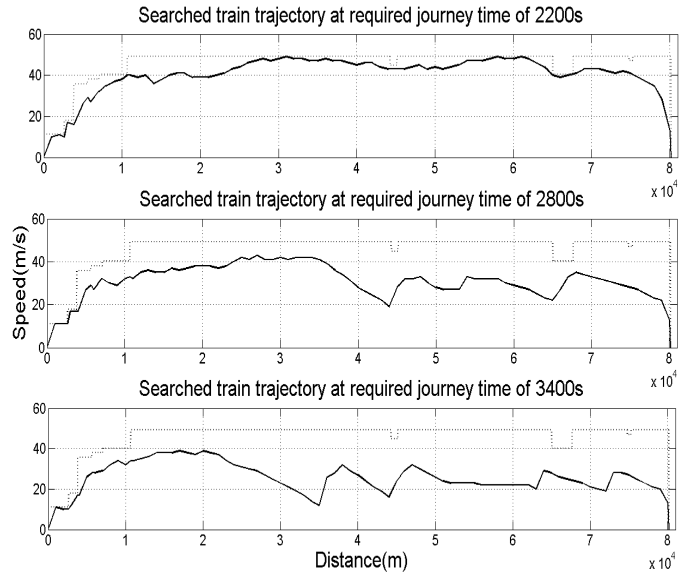


Fig. 6. Optimised journey trajectories using ACO under different journey time conditions.

VI. Results and discussion

The key simulation results are shown in this section. Firstly, trajectories for various scheduled journey time, i.e. 2200 seconds, 2800 seconds, 3400 seconds for ACO, GA and DP are presented in Fig. 6, Fig. 7 and Fig.8.

For journeys with considerable journey time margin, both the ACO and GA algorithms fail to find a smooth trajectory. There is no extended cruising phase and often there is a considerable difference between the maximum and minimum speed 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 2800s.

The journey time cost vs. energy cost curves for different scheduled journey time are compared between the three algorithms. The journey time cost range from 2100 seconds to 3500 seconds with an interval of 100 seconds. These curves are shown

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

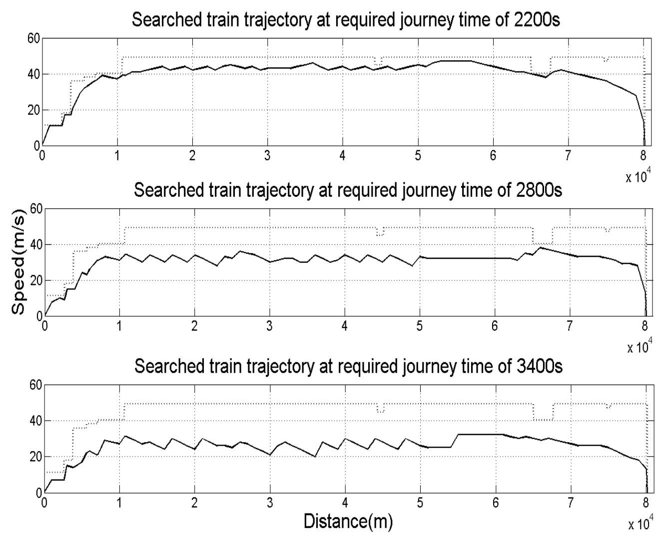


Fig. 8. Optimised journey trajectories using DP under different journey time conditions.

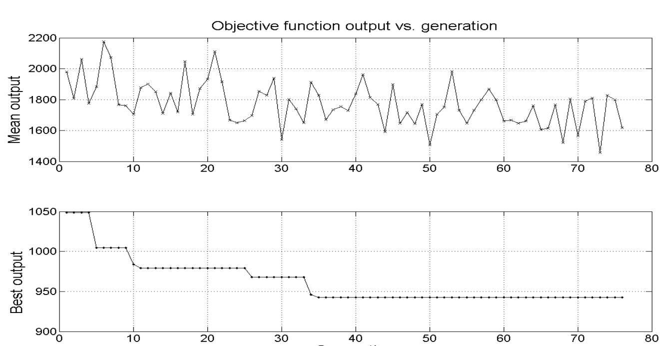


Fig. 9. The best and mean objective function output for journey time of 2800 seconds at each generation for ACO

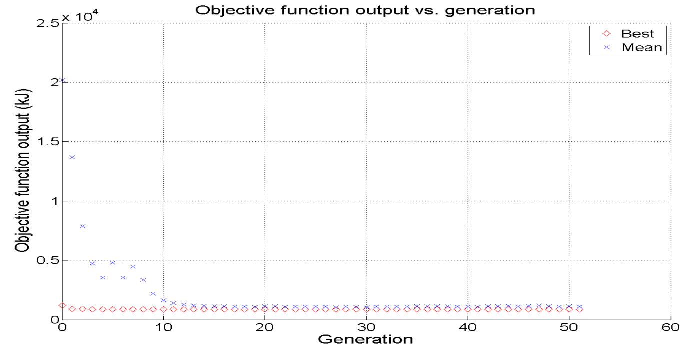


Fig. 10. The best and mean objective function output for journey time of 2800 seconds at each generation for GA

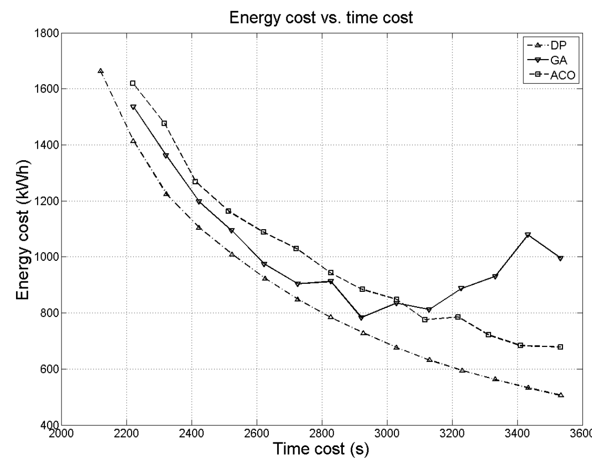


Fig. 11. Journey energy cost vs. time cost curves using different algorithms

TABLE IV

CHARACTERISTIC COMPARISON BETWEEN THREE ALGORITHMS APPLIED ON THE JOURNEY WITH SCHEDULED TIME OF 2800S 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

in Fig.11. Note that each mark in the figure shows a combination of the journey time cost and energy cost for a simulation. 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 seconds using the DP algorithm, the others both failed to converge. At journey times greater than 2800, 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 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.

Dynamic programming 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 algorithm demonstrates its robust searching capability even at even lower journey times, say 2100 seconds. However, the algorithms based on the random Monte-carlo style selection have the possibility of never finding a suitable solution.

Tab. IV shows a comparison of the characteristic between the three algorithms.

VII. Conclusions and Summary

A. Overview

In this paper, methods for single train trajectory optimisation are discussed. Choosing the sequence of control operations is a problem that requires a non-trivial 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 ACO and 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 3 methods have been contrasted and compared. It was found that DP performed better than both GA and ACO. Under certain circumstances 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 1 method should be used to identify optimum trajectories because it is often possible to converge on a solution which is plausible, yet nowhere near optimal.

References

[1] I. Mitchell, "The sustainable railway-use of advisory systems for energy savings," *News view*, no. 151, pp. 2–8, Dec. 2009, technical 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," ser. ICCAS 2002. International Conference on Control, Automation and Systems. IEE, 2001, pp. 394–397.

[3] C. Chang and S. Sim, "Optimising train movements through coast control using genetic algorithms," *Electric Power Applications, IEE Proceeding*, vol. 144, no. 1, pp. 65–73, 1997.

[4] K. K. Wong and 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, Oct 2004.

[5] K. Wong and T. Ho, "Coast control for mass rapid transit railways with searching methods. electric power applications," *Electric Power Applications, IEE Proceedings*, vol. 151, no. 3, pp. 365–376, 2004.

[6] S. Acikbas and M. Soylemez, "Coasting point optimisation for mass rail transit lines using artificial neural networks and genetic algorithms," *Electric Power Applications, IET*, 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," *Electric Power Applications, IET*, vol. 1, no. 5, pp. 675–682, 2007.

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

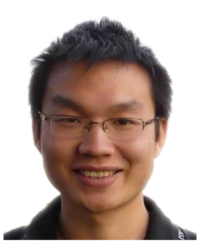
[9] P. Howlett, "The optimal control of a train," *Annals of Operation Research*, vol. 98, pp. 65–87, 2000.

[10] E. Khmelnitsky, "On an optimal control problem of train operation," *IEEE transactions on automatic control*, vol. 45, no. 7, pp. 1257–1266, 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., vol. 15. The Wessex Institute, 2004, pp. 103–112.

[12] S. Effati and H. Roohparvar, "The minimization of the fuel costs in the train transportation," *Applied Mathematics and Computation*, 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 *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 3, 2000, 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 *2009 International forum on information technology and applications*, IEEE. IEEE computer society, 2009, pp. 58–63.
[15] C. Johnson and J. Gibson, "Singular solutions in problems of optimal control," *Automatic Control, IEEE Transactions on*, vol. 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*. McGraw-hill Book Company, 1966.
[17] M. Miyatake and H. Ko, "Optimization of train speed profile for minimum energy consumption," *IEEJ Transactions on Electrical and Electronic Engieering*, vol. 5, no. 3, pp. 263–269, 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 *Service Operations, Logistics, and Informatics (SOLI), 2011 IEEE International Conference on*, Jul. 2011, pp. 589 –594.
[19] I. A. Hansen and J. Pachl, Eds., *Railway timetable & traffic : analysis, modelling, simulation*. Hamburg : Eurailpress, 2008.
[20] S. Lu, S. Hillmansen, and C. Roberts, "A power management strategy for multiple unit railroad vehicles," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 406–420, 2011.
[21] A. Mirabadi and M. Najafi, "Energy management system in hybrid trains using fuzzy control: a comparative study," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 225, no. F3, pp. 267 – 76, 2010.
[22] S. Hillmansen and C. Roberts, "Energy storage devices in hybrid railway vehicles: A kinematic analysis," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 221, no. 1, pp. 135–143, 2007.
[23] M. Dorigo and S. Thomas, *Ant colony optimisation*. The 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," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 226 –235, Jun. 2009.
[25] S. Lu, "Optimising power management strategies for railway traction systems," PhD Thesis, School of Electronic, Electrical and Computer Engineering, University of Birmingham, 2011.
[26] R. E. Bellman, *Dynamic programming*, ser. Rand Corporation research study. Princeton University Press, 1957.



Dr. Shaofeng Lu is a post-doctoral researcher with the Birmingham Centre for Railway Research and Education at the school of Electronics Electrical and Computer Engineering, the University of Birmingham. He received the BEng and PhD degree from the University of Birmingham in 2007 and 2011 respectively. He also has a BEng degree from Huazhong University of Science and Technology, Wuhan, China. All are in Electrical and Electronic Engineering.

His main research area includes railway vehicle traction system modelling, power management strategies, railway electrical network and optimisation algorithms.

Dr. Stuart Hillmansen is a Lecturer in Electrical Energy Systems within the school of Electronic, Electrical and Computer Engineering at the University of Birmingham. He completed a PhD in Imperial College London. His main area of research interest is in hybrid traction systems for use in railway vehicles, and modeling and measurement of energy consumption for railway systems (both AC and DC). He is a member of the Birmingham Centre for Railway Research and Education. He leads the Railway Traction Research Group whose portfolio of activities is supported by the railway industry and government. He has authored a number of papers on railway energy consumption and presented the work at a number of international conferences. He is on the editorial board of the Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit.

Dr. Mark Ho received the B.Eng. and Ph.D. degrees in Electronic and Electrical Engineering at the University of Birmingham, UK, in 1988 and 1994 respectively.

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

Dr. Clive Roberts is Professor in Railway Systems and Director for Research at the University of Birmingham's Centre for Railway Research and Education. His PhD focussed 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, signalling, mechanical interactions and capacity. He currently leads a team of 11 research fellows and 16 PhD students.